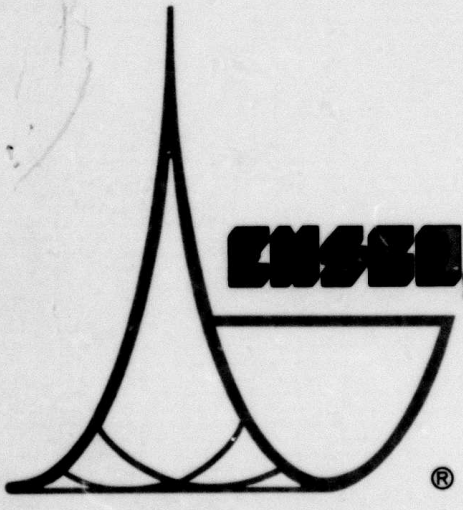
(12) P.5

LEVEL II



ENSCO, INC.

APPROVED FOR PUBLIC RELEASE, DISTRIBUTION UNLIMITED

SAR(01)-TR-79-02
10 October 1979

MANAGEMENT OF A SIGNAL MEASUREMENT DATA BASE (SMDB)

TECHNICAL REPORT NO. 2

PREPARED BY
DONALD L. DIETZ AND JEFFREY S. SHAUB

PREPARED FOR
AIR FORCE TECHNICAL APPLICATIONS CENTER
ALEXANDRIA, VIRGINIA 22314

ENSCO, INC.
SEISMIC APPLIED RESEARCH DIVISION
813 NORTH ROYAL STREET
ALEXANDRIA, VIRGINIA 22314

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER

2. GOVT ACCESSION NO.

3. RECIPIENT'S CATALOG NUMBER

(11 Dec 78-31 Mar 84)

4. TITLE (and Subtitle)

(6) MANAGEMENT OF A SIGNAL MEASUREMENT DATA BASE (SMDB).

5. TYPE OF REPORT & PERIOD COVERED

(9) Technical rept.

6. PERFORMING ORG. REPORT NUMBER

(14) SAR(01)-TR-79-02, TR-2

7. AUTHOR(s)

(10) Donald L. Dietz and Jeffrey S. Shaub

8. CONTRACT OR GRANT NUMBER(s)

(15) F08606-79-C-0014, mw

9. PERFORMING ORGANIZATION NAME AND ADDRESS

ENSCO, INC.
SAR Division
Alexandria, Virginia 22314

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS

ARPA Order-2551

VELA T/9705/B/PMP

11. CONTROLLING OFFICE NAME AND ADDRESS

Advanced Research Projects Agency
Nuclear Monitoring Research Office
Arlington, Virginia 22209

(11)

12. REPORT DATE

10 Oct 79

13. NUMBER OF PAGES

95

14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)

Air Force Technical Applications Center
VELA Seismological Center
Alexandria, Virginia 22314

15. SECURITY CLASS. (of this report)

UNCLASSIFIED

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

(12) 199

APPROVED FOR PUBLIC RELEASE, DISTRIBUTION UNLIMITED

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

ARPA Order No. 2551

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Seismology
Signal Measurements
Event Identification
Expandable Data Base

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This report describes a suite of programs to be used for management of a Signal Measurement Data Base (SMDB). The programs permit a user to write on disk, from tape, files of seismic signal measurements which have been used, for example, in VELA-sponsored identification experiments. The program suite consists of software to establish the SMDB and to enable user interface with the data base. Moreover, the structure of the

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

411669 Dm

20. (continued)

SMDB is also presented here to illustrate its expandable and maintainable format.

This research was supported by the Advanced Research Projects Agency of the Department of Defense and was monitored by AFTAC/VSC, Patrick Air Force Base, FL 32925, under Contract Number F08606-79-C-0014.

AFTAC Project Number:	VELA T/9705/B/PMP
Project Title:	VELA Network and Automatic Processing Research
ARPA Order Number:	2551
Name of Contractor:	ENSCO, Incorporated
Contract Number:	F08606-79-C-0014
Effective Date of Contract:	11 December 1978
Contract Expiration Date:	31 March 1980
Project Manager:	Theodore J. Cohen (703) 548-8666

SUMMARY

This report describes a suite of programs to be used for management of a Signal Measurement Data Base (SMDB). The programs permit a user to write on disk, from tape, files of seismic signal measurements which have been used, for example, in VELA-sponsored identification experiments. The program suite consists of software to establish the SMDB and to enable user interface with the data base. Moreover, the structure of the SMDB is also presented here to illustrate its expandable and maintainable format.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

SECTION I
INTRODUCTION

The purpose of the Data Base Transfer (Task 4.3.1 under Contract Number F08606-79-C-0014) is to transfer the Event Discriminant Data Base from the PDP-15/50 to the PDP-11/70, and to do so under the following constraints:

- Preserve the signal measurements for all event-stations processed under Contract Number F08606-79-C-0014.
- Establish an expandable and maintainable Signal Measurement Data Base (SMDB) on the PDP-11/70 that:
  - satisfies the first constraint above
  - may be updated with additional signal measurements for event-station data processed by other contractors.
- Provide FORTRAN-compatible software utilities to:
  - initialize the SMDB Directory and Free-Block File
  - update the SMDB by event, station, measurement, or contractor
  - access the SMDB for information by event, station, measurement, or contractor.
- Write selected driving programs to demonstrate an:
  - SMDB update with signal measurements discussed in the first criterion
  - SMDB access to list Directory information and signal measurements

- SMDB access to compute unbiased network averages of signal measurements for a specified event.

- Demonstrate this software contingent on the availability of UNIX operating system utilities provided by the Government.

Section II of the report describes the detailed structure of the SMDB. Section III discusses the establishment of the SMDB, utilizing signal measurements for all event-station data processed under Contract Number F08606-79-C-0014. Section IV presents conclusions and recommendations for future work. Finally, documentation of the software utilities developed during the task is provided in Appendix A of this report, while listings of the software may be found in Appendix B. Appendix C contains a description of ENSCO's raw signal measurement tape format.

SECTION II
STRUCTURE OF THE SIGNAL MEASUREMENT DATA BASE (SMDB)

The Signal Measurement Data Base (SMDB) uses a multi-file structure which is designed and implemented in FORTRAN IV for the PDP-11/70. The principal items comprising the SMDB are vectors of signal measurements obtained by preprocessing waveform data from various recording stations. The design of the data base was predicated on the following criteria:

- Expandability with respect to events, stations, and measurements.
- Allocations for data from up to four contractors.

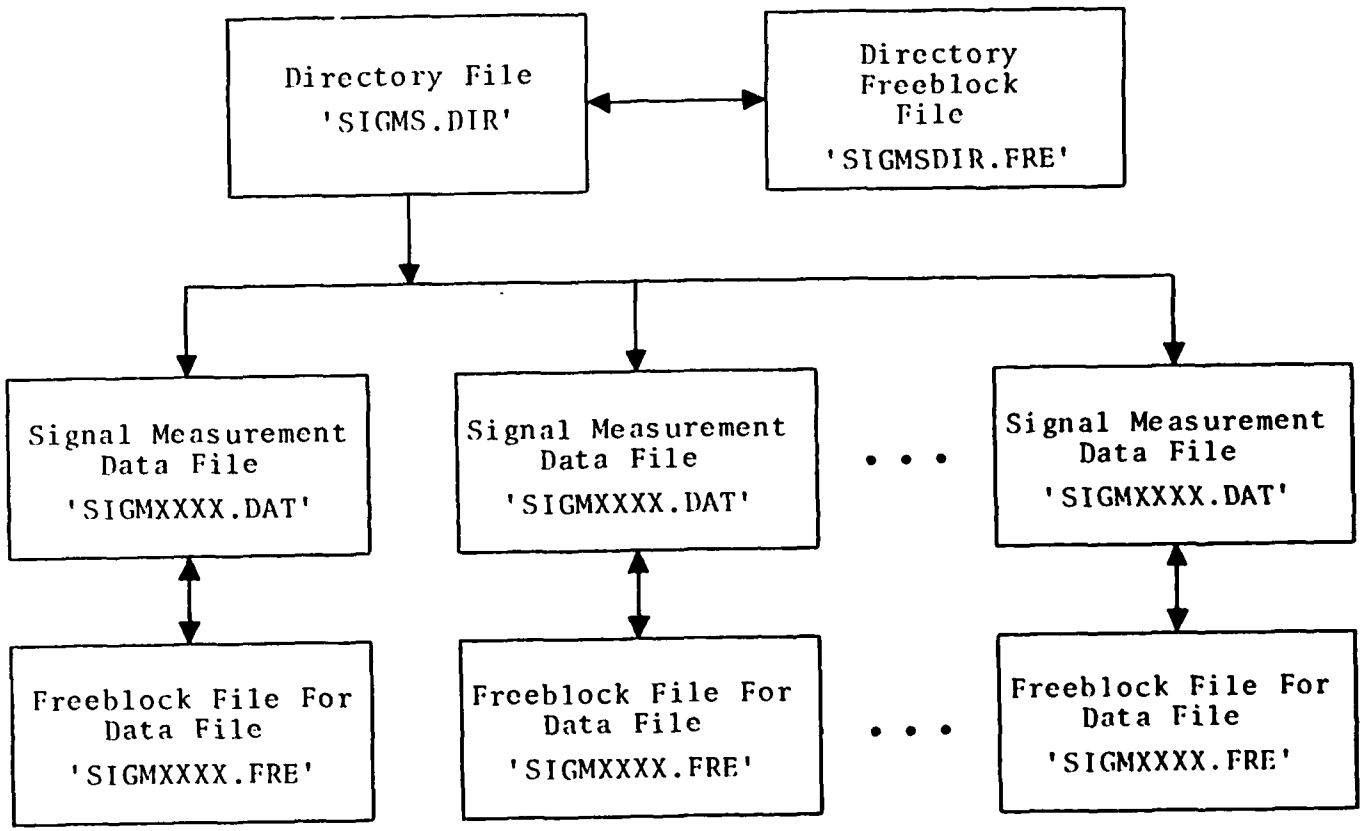
To this end, a linklist structure was employed in organizing the data.

The SMDB consists of four types of files:

- A Directory File (SIGMS.DIR).
- A Directory Freeblock File (SIGMSDIR.FRE).
- Signal Measurement Data Files (one file for each event).
- Freeblock File for Data Files (one file for each event).

The general organization of the SMDB is shown in Figure II-1.

The directory file, SIGMS.DIR, consists of a header record followed by Event Base Vectors and Station Entry Vectors. These vectors are for sorting and retrieving event/



Note: A signal measurement data file and its associated freeblock file exist for each event. File names are as shown with the event designation number substituted for the characters XXXX.

FIGURE II-1

ORGANIZATION OF THE SIGNAL MEASUREMENT DATA BASE (SMDB)

signal information. The header record contains the following information:

- The current number of events in the system.
- The record number of the first event entry.
- The current number of directory records that are in use.
- The maximum number of directory records allowed.

All entries are allocated four bytes of storage.

The structure of the Event Base Vector and Station Entry Vector are shown in Figure II-2. Each entry record is 128 bytes long and consists of thirty-two four-byte words. The attributes of the Event Base Vector are defined as follows:

- Event Sequence Number - A virtual integer J, where $1 \leq J \leq$ # of events, which indicates the order an event appears in the directory. For example, the fifth event listed in the directory has an event sequence number of 5. The sequence number for an event changes when the SMDB is updated with new events.
- Event Designation Number - A unique four-digit number appearing as the last four characters of the ENSCO-alphanumeric event designation.
- Event Origin Time - Source time in YY DDD HH MM SS (5 word integer format).
- Event Latitude - Source latitude in degrees, N(+), S(-).
- Event Longitude - Source longitude in degrees, E(+), W(-).
- Event m_b - Event bodywave magnitude.

EVENT BASE VECTOR

Event Sequence Number (Virtual)	Event Designation Number	Event Origin Time	Event Latitude $+N^{O}$	Event Longitude $+E^{O}$	Event m_b	Event Depth	Unused (22*4 words)	Backward Pointer To Last Event	Forward Pointer To Next Event	Total Number Of Stations	Pointer To First Station

STATION ENTRY VECTOR

Station Name	Station Number	Unused (15*4 words)	Backward Pointer To Last Station	Forward Pointer To Next Station	Total Number Of Contractors	Contractor Name	First Measurement Record	Number Of Measurements	. . .	. . .	. . .

FIGURE II-2
DIRECTORY ENTRIES

- Event Depth - Source depth in km.
- Backward Pointer to Last Event - The record number corresponding to the previous event base vector (zero if first event).
- Forward Pointer to Next Event - The record number corresponding to the next event base vector (zero if last event).
- Total Number of Stations - The total number of station entry vectors for the event. This corresponds to the total number of unique stations for which measurements are available for a given event.
- Point to First Station - The record number corresponding to the first station entry vector for the event.

The attributes of the Station Entry Vector are defined as follows:

- Station Name - The four-character ENSCO-alphanumeric station designation.
- Station Number - The integer number K, $1 \leq K \leq 50$, assigned to a given station.
- Backward Pointer to Last Station - The record number corresponding to the last station entry vector (zero if first station).
- Forward Pointer to Next Station - The record number corresponding to the next station entry vector (zero if last station).
- Total Number of Contractors - The total number of contractors contributing signal measurements for an event-station (maximum of four).

- Contractor Name - The four-character alphanumeric designation for a contributor to the data base.
- First Measurement Record - The record number in the signal measurement data file corresponding to the first record of signal measurements.
- Number of Measurements - The total number of measurement values contributed by the associated contractor.

Both the event base vectors and the station entry vectors have unused locations where, if desired, additional information can be stored.

The directory freeblock file, SIGMSDIR.FRE, consists of a header record containing the current number of non-sequential free records in the directory file. This header record is followed by records consisting of 256 four-byte words. These records each contain up to 256 record numbers. The function of this freeblock is to allow the reuse of records freed by the deletion of entries in the directory file. This practice minimizes the size of the directory file.

Each event has one signal measurement file with a name of the form SIGMXXXX.DAT, where XXXX is the event designation number. The records in these files consist of 64 four-byte words. Two types of records make up the data file: a header record and data records (multiple). The header record contains the current number of data records in use and the maximum number of data records allowed in the first two positions. The structure of the data records is shown in Figure II-3. The major attributes are defined as follows:

31 Measurement Values (4 * byte word)	31 Measurement Labels (4 * byte word)	Last Measurement Record	Next Measurement Record

(file name of form 'SIGMXXXX.DAT' where XXXX=Event Designation Number)

FIGURE II-3

STRUCTURE OF DATA RECORD IN SIGNAL MEASUREMENT DATA FILE

- Measurement Value - A signal parameter obtained by preprocessing waveform data from various recording stations. In the present case, the measurements are ordered into four categories depending on the type of data from which the discriminants are derived (namely, long-period signals/noise, short-period regional signals/noise, short-period teleseismic signals/noise, and short-period signals/noise).
- Measurement Labels - A four-character alphanumeric designation associated with each measurement value. This label is used as a keyword in locating a given signal parameter.
- Last Measurement Record - The record number corresponding to the last data record associated with a given contractor (zero if first record).
- Next Measurement Record - The record number corresponding to the next data record associated with a given contractor (zero if last record).

Each event also has a freeblock file associated with its signal measurement file. The files are named analogously to the data file. The file name is of the form SIGMXXXX.FRE, where XXXX is the event designation number. The structure and function of this file are the same as those of the directory freeblock file.

SECTION III
ESTABLISHMENT OF THE SMDB

The SMDB is established by executing DDBASE. This routine takes a labeled 1600 bpi 9-track signal measurement tape as input. The data headers for each event-station file should follow the description in Table III-1*. Before DDBASE is initially run, the program INITIAL must be executed. This program initializes the directory files of the data base and deletes any data files that may be present. In general, the UNIX command SMT must also be performed with the appropriate options before any execution of DDBASE. These routines along with other support routines and subroutines are documented in Appendix A. The source file names and the names of the associated source files and libraries (if any) are also provided. The algorithms can be executed by following the standard UNIX FORTRAN guidelines. Appendix B contains compiled listings of the data base management and demonstration programs.

---

* The conversion of the data header from the standard ENSCO format (see Appendix C) is performed by the IBM 360/44 program COPY99.

TABLE III-1

SIGNAL MEASUREMENT DATA HEADER
(PAGE 1 OF 3)

Position	Data Type	Description
1	I	Seismogram number (not used).
2	I	Number of components (always equal to one).
3	I	Edit length (always equal to 100).
4	F	Sample rate (not used).
5	F	Edit start time: hours
6	F	Edit start time: minutes } (not used).
7	F	Edit start time: seconds
8-9	A	'TIHEADER'.
10-12	A	Event designation (word 12 is the event designation number).
13-14	A	Data type ('DISCR').
15	A	Data orientation code ('RAW').
16-47	A	Site status table (not used).
48	A	'bbYY'
49	A	'bDDD'
50	A	'bbHH' } event origin time.
51	A	'bbMM'
52	A	'bbSS'
53	A	Confidence of source time (PDE code) (not used).
54-55	A	Event latitude ($\pm 90^{\circ}$N) (represents F8.3).
56-57	A	Event longitude ($\pm 180^{\circ}$E) (represents F8.3).
58	A	Event depth (represent integer).
59	A	Event m_b (represents F4.2).

TABLE III-1

SIGNAL MEASUREMENT DATA HEADER
(PAGE 2 OF 3)

Position	Data Type	Description
60	A	Event M_s (represents F4.2).
61	A	Event m_b (represents F4.2).
62	A	ENSCO estimated M_s (represents F4.2).
63-64	A	Information source (e.g., 'NEIL').
65	A	Event-station tectonic class code.
66	A	Not used.
67	A	Site number (represents integer).
68-69 • • •	A • • •	Signal measurement fields for specific phase (represents F8.3) (meaningless in context of signal measurement tape).
96-97	A	Signal measurement fields for specific phase (represents F8.3) (meaningless in context of signal measurement tape).
98	A	Array name ('SRObb').
99-157	A	Not used.
158	A	Array name.
159-160	A	Site latitude ($\pm 90^{\circ}$N) (represents F8.3).
161-162	A	Site longitude ($\pm 180^{\circ}$E) (represents F8.3).
163	A	P-wave arrival seconds into edit (represents integer).
164	A	S-wave arrival seconds into edit (represents integer).
165	A	LQ-wave arrival seconds into edit (represents integer).
166	A	LR-wave arrival seconds into edit (represents integer).

TABLE III-1

SIGNAL MEASUREMENT DATA HEADER
(PAGE 3 OF 3)

Position	Data Type	Description
167	A	Estimated LR length (seconds) (represents integer).
168-169	A	Station azimuth (degrees) (great circle beam direction) (represents F8.3).
170	A	Station elevation (meters) (great circle beam direction) (represents integer).
171-172	A	Source-to-station epicentral distance great circle distance (degrees) (represents F8.3).
173-322	A	Not used.
323-324 • • •	A • • •	Signal measurement fields for specific phases (represents F8.3) (meaningless in context of signal measurement tape).
373-374	A	Signal measurement fields for specific phases (represents F8.3) (meaningless in contest of signal measurement tape).
375	A	Not used.

SECTION IV
CONCLUSIONS AND RECOMMENDATIONS

An expandable and maintainable Signal Measurement Data Base (SMDB) program suite has been designed and compiled on the PDP-11/70 under the UNIX operating system. Tapes (from the SDAC) which were used in the VELA-sponsored identification experiment, and which were used in the development and test of the SMDB program suite, are:

1. L22882 (IBM 360/44)
2. L13033 (DEC PDP-15/50).

These tapes can be used in future efforts to refine the SMDB program suite.

It should be noted that several problems were encountered with the UNIX FORTRAN compilers during attempts to implement the SMDB routines. Specifically, the routines were initially coded in a modified version of DEC RT-11 FORTRAN IV (Rottman, 1975). (The UNIX command to request this compiler is FORTRAN.) Difficulties encountered with this compiler included problems which are apparently related to array size and to the argument lists. That is, a routine could be made to generate routine errors, or not to do so, by increasing or decreasing the size of its arrays, respectively. Associated error messages, in general, indicated a bus error or a segmentation violation. Regardless, a solution to this problem was not readily apparent. Further, the number of

arguments in a subroutine argument list appeared to be limited to eight. Any additional arguments were not passed. It was possible to circumvent this problem by passing arguments in a named common block.

Toward the end of this task, a new FORTRAN compiler was provided (Anon., 1978). This compiler is based on American National Standard (ANS) FORTRAN 77. (The UNIX command to request this compiler is F77.) An investigation was made into the feasibility of converting the SMDB routines to the FORTRAN 77 standard. Necessary changes would include substitution of the ENCODE/DECODE statements with equivalent statements, modification of the file I/O, modification of character variables, and modification of the code to allow for irregularities in the new compiler. ENCODE/DECODE statements can be replaced with 'reads' and 'writes' to internal files. This replacement is not possible on a one-to-one basis, and some additional program structuring will be necessary to provide for parallel operations. The file I/O initiated using the routines SETFIL and DFILE does not appear to be equivalent to using SETFIL and DEFINE FILE in the original compiler. In particular, it seems that the file to be accessed must already exist in order to be accessed (i.e., the system does not create a new file). This was not the case with the compiler called using FORTRAN. Another difference in file I/O requirements was that the last record written in a file could not be read subsequent to the write to the file. This difficulty may indicate that it is necessary to close and open the file between write and read operations. An example of one irregularity encountered during the investigation of the F77 compiler is the failure to

compile a logical expression with a unary minus (e.g., if (IFLG.EQ.-1) go to 100). The same logical expression without the minus sign, however, does compile.

The results of the investigation made indicate that the F77 compiler is sufficiently different from the FORTRAN compiler as to require a significant effort to convert the SMDB from the FORTRAN to the F77 compiler. It will first be necessary to establish the operating characteristics of the new compiler (i.e., the logic sequences necessary to accomplish a given operation). After the methods for performing the desired operations have been established (usually done by the trial-and-error method), the SMDB routines can be restructured (i.e., converted).

At this time, a competent FORTRAN language compiler is not available on the SDAC PDP-11/70. Note, however, that the use of the FORTRAN language is not recommended by the designers of the UNIX operating system for major applications using UNIX.

As a final comment, several additional routines could be added to make the SMDB more versatile. These include a function to delete event and/or station entries and a function to modify individual measurements resident in the data base. The design of the SMDB included consideration of these avenues for development. Several of the routines documented in Appendix A could be used to implement these options.

SECTION V
REFERENCES

Anon., 1978; American National Standard Programming Language FORTRAN, ANSI X3.9-1978, American National Standards Institute, 1430 Broadway, New York, NY.

Kerningham, B., undated; UNIX For Beginners, Bell Laboratories, Murry Hill, NJ.

Ringdal, F., 1975; Maximum Likelihood Estimation of Seismic Event Magnitudes from Network Data, Technical Report No. 1, Texas Instruments Report No. ALEX(01)-TR-75-01, AFTAC Contract Number F08606-75-C-0029, Texas Instruments Incorporated, Dallas, TX.

Rottman, J. N., 1975 (revised); A Guide to the UNIX FORTRAN System, Princeton University, Princeton, NJ.

Thompson, K., and D. M. Ritchie, 1975; UNIX Programmer's Manual, Bell Telephone Laboratories, Incorporated, Murry Hill, NJ.

APPENDIX A
DOCUMENTATION OF SIGNAL MEASUREMENT DATA BASE (SMDB) ROUTINES

This appendix contains descriptions of the main routines and associated subroutines used to manage and demonstrate the SMDB. These descriptions include definitions of arguments, commons, and data types. Comments on the use of the routines are also provided. Applicable source files resident on the PDP-11/70 system are listed.

These routines can be incorporated into additional algorithms to manipulate the SMDB. Several examples of this type of application are generation of files of unbiased network magnitude measurements, computation of discriminants, and classification of events.

Subroutine:

ACCESS

Arguments:

(NDSG, STAT, CONTR, MSLAB, NRET, RMEAS, RLAB, IEVR).
NDSG - integer * 4
STAT - real * 4
CONTR - real * 4
MSLAB - real * 4
NRET - integer * 4
RMEAS - real * 4 array of length $\geq$ NRET
RLAB - real * 4 array of length $\geq$ NRET
IEVR - integer * 4

Commons:

COMMON/FLAGS/IFLGMS, IFLGEV
IFLGMS - integer * 2
IFLGEN - integer * 2
COMMON/ASVAR/NXREC1, NXREC2, NXREC3, NXREC4

Description:

ACCESS accesses the signal measurement data base and returns the measurements.

NDSG is an event designation number.

STAT is a four-character station name (if it is ****, the given measurement is returned for all stations reporting the event).

CONTR is a four-character contractor name.

MSLAB is a four-character measurement label.

NRET is the number of measurements returned.

RMEAS contains the measurements to be returned.

RLAB contains the measurement labels when IFLGMS=0.

IEVR is the event record number in the directory file SIGMS.DIR.

IFLGMS = { 0, returns all measurements for the given station (in RMEAS) with the associated labels (in RLAB) — WARNING: it should be noted that there is no guarantee that all of the measurements will be in order (or exist) across stations.
1, return only indicated measurement for the given station.

IFLGEV = { 0, search for event
1, use event record number in IEVR.

NXREC1, NXREC2, NXREC3, NXREC4 are the associated variables for the maximum of four data base files open at one time.

Restrictions:

1) The event data file is opened on logical unit number 2.
2) It is assumed that the directory file SIGMS.DIR is already open on logical unit number 1.

Source File:

access.for

Associated Source File:

search.for

Subroutine:

CHKFRE

Arguments:

(LUNIT, IFLAG, NFB, IFREC).
LUNIT - integer * 2
IFLAG - integer * 2
NFB - integer * 4
IFREC - integer * 4

Commons:

COMMON/ASVAR/NXREC1, NXREC2, NXREC3, NXREC4
all variables integer * 2

Description:

CHKFRE checks the indicated freeblock file and either returns a free record number of extends the freeblock file. The file is modified accordingly and the total number of freeblocks before the call is returned.

LUNIT is the logical unit number associated with the freeblock file (in practice LUNIT=3 is the freeblock file for the directory and LUNIT=4 is the freeblock file for the event data file.

IFLAG { 0, returns free record number in IFREC
1, freeblock extended with record number in IFREC.

NFB is the number of freeblocks prior to a call.

IFREC is the free record number.

NXREC1, NXREC2, NXREC3, NXREC4 are the associated variables for all possible open data base files.

Restrictions:

1) The appropriate freeblock files must be open on logical unit LUNIT.

Source File:

chkfre.for

Main Routine:

COPY99

Input:

Signal measurement tape in standard ENSCO format.

Description:

COPY99 is an IBM 360/44 program used to convert the data headers of the signal measurement tape to the format expected by the PDP-11/70 program DDBASE and output the results to another tape. The output tape is input to the signal measurement data base management routines. Prior to execution of this program, the character string 'NSKIP' should be changed to the number of data files to skip on the input tape before starting the conversion and copy. The input tape is accessed on unit 8 and the output tape is accessed on unit 9.

Main Routine:

DDBASE

Input:

9-track signal measurement tape with appropriate EBCDIC data headers on logical unit 0.

Description:

DDBASE accesses the information on a labeled 1600 bpi 9-track signal measurement tape and performs the necessary conversions and formating before providing the information to sub routine UPDATE. This routine includes a look-up table for station names based on the station number, a look-up table for measurement labels, EBCDIC to ASCII conversion, and IBM 360/44 to PDP-11/70 floating-point conversion.

Restrictions:

1) The 9-track input tape must have the correct EBCDIC header.
2) All information for an event on a tape must be consecutive.
3) UNIX command smt must be performed before execution of DDBASE with the appropriate options.
4) The algorithm currently requires the tape to be mounted on unit 0 (allowable units are 0-7).

Source File:

ddbase.for

Associated Source Files:

update.for
search.for
chkfre.for

Program Source File:

ddbasep.for

Associated libraries (in addition to normal FORTRAN library):

/lib/f4u

Main Routine:

EVLIST

Input: (from terminal)

Logic flag (IFLG) (requested inputs 0 or 1) range of event sequence numbers (virtual).

Description:

EVLIST displays the event-station information stored in the data base in the form requested by the input parameters.

If a zero input for the logic flag the event information and the stations reporting are displayed.

If a one is input the individual station measurements for contractor ENSC are included.

Source File:

evlist.for

Associated Source Files:

search.for
access.for

Program Source File:

evlistp.for

Main Routine:

INITIAL

Description:

INITIAL initializes the signal measurement data base. This is accomplished by deleting all existing event data files (if any) and by initializing the header records of the directory file (SIGMS.DIR) and its freeblock file (SIGMSDIR.FRE).

Program Source File:

initial.for

Subroutine:

MBIAS

Arguments:

(RMX, NSTA, NDISC, DISCR).

RMX - real * 4 of length $\geq$ NSTA * NDISC.

NSTA - scalar (integer * 2).

NDISC - scalar (integer * 2).

DISCR - real * 4 array of length $\geq$ NDISC.

Description:

MBIAS performs Ringdal's maximum likelihood estimation of event magnitude (Ringdal, 1975) using signal and noise measurements.

RMX is a matrix whose elements contain signal or noise measurements, or a flag indicating that the element is to be ignored.

$$RMX_{ij} = \begin{cases} \text{signal measurement} + 1000 \\ \text{noise measurement} - 1000 \\ 0 \text{ if it is to be ignored.} \end{cases}$$

NSTA is the number of stations.

NDISC is the number of measurements per station.

DISCR contains the unbiased magnitude estimates.

Restrictions:

1) RMS contains the measurements for station 1 followed by the measurements for station 2, etc. until the measurements for station NSTA have been stored.

2) NSTA must be ≤ 50.

Source File:

mbiascat.for

Main Routine:

NAVE and (NAVE2)

Input: (from terminal)

Event designation number.

Description:

NAVE and NAVE2 are drivers to demonstrate the subroutine MBIAS. The unbiased network averages are computed for several measurements for the designated event.

Source File:

nave.for and nave2.for

Associated Source Files:

mbiascat.for
search.for
access.for

Program Source File:

navep.for

Main Routine:

RETURN

Input: (from terminal)

Event designation number
Station name
Measurement label.

Description:

RETURN is a debugging routine to test subroutines ACCESS and SEARCH. Various measurements are returned depending on the input parameters and the setting of flags hardwired into the program.

Source File:

return.for

Associated Source Files:

search.for
access.for

Program Source File:

returnp.for

Subroutine:

SEARCH

Arguments:

(IFLAG, KEY, EVREC, STREC, ICON, MSREC, MSIND, ISRCH)
IFLAG - integer * 2
KEY - four word real * 4 array
EVREC - integer * 4
STREC - integer * 4
ICON - integer * 4
MSREC - integer * 4
MSIND - integer * 2
ISRCH - integer * 2

Commons:

COMMON/ASVAR/NXREC2, NXREC2, NXREC3, NXREC4
all variables INTEGER * 2.

Description:

SEARCH searches the signal measurement data base for a match to a specific keyword and returns the appropriate record number and index (if applicable).

IFLAG =
- 0, full search for a given measurement (needs all keywords)
- 1, search for event designation number
- 2, search for station name (needs evrec)
- 3, search for contractor name (needs strec)
- 4, search for measurement (needs icon) (returns record number in 'sigmxxxx.det' file and the index into the record).

KEY is the four-word array containing the keywords

KEY(1) = event designation number (should be converted to floating point prior to call)

KEY(2) = four-character station name

KEY(3) = four-character contractor name

KEY(4) = four-character measurement label.

EVREC is the event record number in the directory file, SIGMS.DIR.

STREC is the station record number in the directory file, SIGMS.DIR.

ICON is the index into the station record of the contractor block - possible values are 21, 24, 27, and 30.

MSREC is the measurement record number in the event data file SIGMXXX.DAT.

MSIND is the index into the measurement record of the measurement value

ISRCH is an error flag

$$ISRCH = \begin{cases} 0, & \text{successful search} \\ 1, & \text{unsuccessful search} \end{cases}$$

NXREC1, NXREC2, NXREC3, NXREC4 are the associated variables of all possible open data base files.

Restrictions:

1) If EVREC or STREC are equal to zero when IFLAG=2 or 3, respectively, the program is stopped. This also occurs if the number of stations or measurements is zero.

2) The content of KEY(1) must be in floating point representation.

3) The directory file SIGMS.DIR must be open on logical unit number 1 and the event data file must be open on logical unit number 2.

Source File:

search.for

Subroutine:

UPDATE

Arguments:

(IERR)
IERR - integer * 2

Commons:

COMMON/ARG/ESINFO
ESINFO - 141-word real * 4 array
common/ASVAR/NXREC1, NXREC2, NXREC3, NXREC4
all variables integer * 2

Description:

UPDATE establishes the signal measurement data base given the pertinent event-station information and flags contained in array ESINFO.

IERR is an error flag.

IERR = {0, no errors
1, error in establishing data base.

ESINFO is the information packet passed to this data base routine.

ESINFO(1) = event designation number (integer * 4).
ESINFO(2) = bbYY
ESINFO(3) = bDDD
ESINFO(4) = bbHH
ESINFO(5) = bbMM
ESINFO(6) = bbSS
} event origin time (integer * 4).
ESINFO(7) = event latitude ($\pm 90^{o}$N)(represented as F8.3 in this data base).

ESINFO(8) = event longitude (+180°E) (represented as F8.3 in this data base).

ESINFO(9) = event magnitude (m_b) (represented as F4.2 in this data base).

ESINFO(10) = depth of event (integer * 4).

ESINFO(11) = event flag IFLGEV (integer * 4).

ESINFO(12) = four-character station name.

ESINFO(13) = station number (integer * 4).

ESINFO(14) = station flag IFLGST (integer * 4).

ESINFO(15) = four-character contractor name.

ESINFO(16+(K-1)*2) = four-character measurement label K=1,62.

ESINFO(17+(K-1)*2) = associated measurement value K=1.62.

ESINFO(140) = event record number in the directory file SIGMS.DIR (integer * 4).

ESINFO(141) = station record number in the directory file SIGMS.DIR (integer * 4).

IFLGEV (ESINFO(11)) is a flag that controls the logic flow of this routine in regard to events.

$$\text{IFLGEV} = \begin{cases} 1, & \text{search for event} \\ 0, & \text{add new event} \\ -1, & \text{update previous event (i.e., use event record number supplied in ESINFO(140)).} \end{cases}$$

IFLGST (ESINFO(14)) is a flag that controls the logic flow of this routine in regard to stations.

$$\text{IFLGST} = \begin{cases} 1, & \text{search for station} \\ 0, & \text{add new station} \\ -1, & \text{update previous station (i.e., use station record provided in ESINFO(141)).} \end{cases}$$

The following files are associated with the indicated logical units.

SIGMS.DIR (directory file) logical unit = 1.

SIGMXXXX.DAT (event data file, where XXXX is the event designation number) logical unit = 2.

SIGMSDIR.FRE (directory freeblock file) logical unit = 3.

SIGMXXXX.FRE (freeblock file for event data file) logical unit = 4.

NXREC1, NXREC2, NXREC3, NXREC4 are the associated variables for all possible open files.

Comments:

1) If an event is either flagged as new (i.e., not presently in the data base) or not found in a search, the station flag is set to zero to indicate that the stations are also new.
2) For the most efficient execution of this routine, the event-station information for a given event should be provided consecutively during a given run.
3) Update is written in such a way that measurements associated with an existing label can be replaced by the value in the information packet.
4) In general, the data base structure implemented by this routine is expandable with respect to events, stations, and measurements (linklist structure).

Restrictions:

1) The directory file must be open on logical unit 1 and the directory freeblock file must be open on logical unit 3.
2) Logical units 2 and 4 must be free.

3) The number of associated measurement labels and values must be ≤ 62. If the number is less than 62, the last measurement value must be followed by a word containing four blank characters.

Source File:

update.for

Associated Source Files:

search.for
chkfre.for

APPENDIX B
LISTINGS OF SIGNAL MEASUREMENT DATA BASE (SMDB) PROGRAMS

This appendix contains compiled listings of various data base management and demonstration programs. They are COPY99, INITIAL, DDBASE, EVLIST, NAVE, and RETURN. A modified version of the main routine NAVE is also included as NAVE2.

```
/*  7384  USER=WELTMAN   FILE=NCBN.DIRIZ.EGCPY99         14:10 CC1  3
//COPY99 EXEC FORTRAN(MAF)

FORTRAN IV    MODEL 44  PS      VERSION 3,  LEVEL 3   DATE  WED OCT  3, 1979   AT 14.12    PAGE  0001

 0001             DIMENSION HDBC(375),IHDBC(375),IHDRI(375),HDRI(375),DATA(4111)
 0002             EQUIVALENCE (HDFO,IHDBC,DATA),(DATA(2),TAPENO),(IHDRI,HDRI)
 0003             REAL*4 ARRAY(14)/'ILPARSPSAIFANCRSSNC KS3CDBT4',7*0./
 0004             INTEGER*4 ECITEI(20)/'ECIO',19*'    '/
 0005             LOGICAL*1 TAPENO(6)
 0006             DATA NARAY/7/
           C
           C         THIS ROUTINE CONVERTS FROM NEW-FORMAT EVENT HEADERS TO THE
           C         ALPHANUMERIC HEADER EXPECTED BY THE PDP-11 SIGNAL SEAS.
           C         INITIALIZATION PROGRAM. (THIS ROUTINE CAN STILL CONVERT EDIT
           C         TAPES)----THE OUTPUT TAPE MUST BE LABELED (RUN ENSCO.GENTAPE)
           C
           C         HDSI:  360/44 HEADER INPUT TO HDRIO
           C         HDBC:  PDP-11 HEADER OUTPUT BY HDRIO
           C         ARRAY: 4 CHARACTER ARRAY NAMES
           C         NARAY: NUMBER OF ARRAY DESIGNATIONS ( <15 )
           C
 0007             CALL TITAPE('RING',9,X,X,IC)
 0008             IF (IC.NE.0) GO TO 80
 0009             CALL TITAPE('REWI',9,X,X,IC)
 0010             CALL STATUS('WAIT',9,IC)
 0011             CALL STATUS('CLEA',9,IC)
 0012             CALL NOINTE ('COIPO-1','CCNT',9,HDFC,NSKP,IC)
 0013             IF (IC.NE.0) GO TO 80
           C      OPEN INPUT TAPE
 0014             CALL TICPEN(8,HDEC,680)
 0015             WRITE (6,45) TAPENO
 0016          45 FORMAT('1FILES COPIED FROM TAPE ',6A1/1HO)
 0017             J=NSKIP
 0018             IF (J.EQ.0) GO TO 20
 0019             DO 15 I=1,J
 0020          12 CALL TITAPE('FSPI',8,X,X,IC)
 0021             CALL STATUS('CHEC',8,IC)
 0022             IF (IC) 12,15,12
 0023          15 CONTINUE
 0024          20 CALL TIREAD(8,HDRI,1500,IC,690,690,680)
 0025             J=J+1
 0026             WRITE (6,25) J,(HDRI(I),I=10,12)
 0027          25 FORMAT(I5,4X,3A4)
 0028             CALL MCVEBT('    ',HDBC(8),4)
 0029             CALL MCVE(HDBC(8),HDBC(9),367)
           C
           C         COPY SUBSET HEADER FIELD: SEISMOGRAM NO., NO. OF
           C         COMPONENTS, EDIT LENGTH(POINTS), SAMPLE RATE, EDIT START
           C         IN HRS, MINS, AND DECISECONDS
           C
 0030             CALL MCVE(HDRI,HDPC,7)
           C
           C         SET DATA SPECIFICATION FIELD: EVENT DESIGNATION, DATA TYPE,
           C         DATA ORIENTATION CODE, UP STATUS FOR GIVEN ARRAY DESIGNATION,
           C         BLANKS IN EXPANSION FIELD
           C
 0031             CALL MCVEBT('TIPEADER',HDBC(8),8)
 0032             CALL MCVE(HDRI(10),HDBC(10),5)
 0033             HDFO(15)=HDRI(22)
 0034             DO 32 I=1,NARAY
 0035             IF (HDRI(8).EQ.ARRAY(I)) GO TO 34
 0036          32 CONTINUE
 0037             I=15
 0038          34 CALL MCVEBT('   1',HDBO(16),4)
 0039             CALL MCVE(HDFC(16),HDFC(17),14)
 0040             CALL MCVEBT('   0',HDFC(15+I),4)
 0041             ICPP=15*(I-1)
           C
           C         CONVERT SOURCE INFORMATION FIELD: SOURCE TIME FROM INTEGER
           C         WORDS YEAR-DAY AND SECONDS INTO DAY, TO ALPHA STRING
           C         ' YY DDD  HH  MM  SS', CONFIDENCE OF SOURCE TIME, LATITUDE,
           C         LONGITUDE, DEPTH, MB, MS, OTHER MB, OTHER MS, INFORMATION
           C         SOURCE, REGION AND COMMENTS, BLANKS IN EXPANSION FIELD
           C
 0042             IYRDA=IHDRI(53)
 0043             ISECD=IHDRI(52)
 0044             IYR=IYRDA/1000
 0045             IDA=IYRDA-1000*IYR
 0046             IHR=ISECD/3600
 0047             IMIN=(ISECD-3600*IHR)/60
 0048             ISEC=ISECD-3600*IHR-60*IMIN
```

COPY99
(PAGE 2 OF 2)

```
FORTRAN IV    MODEL 44   FS      VERSION 3, LEVEL 3   DATE  WED OCT  3, 1979   AT 14.12    PAGE  0002

0049              CALL RCLLER(IYR,DUMY,DUMY,MDRC(48),DUMY)                                  00000395
0050              CALL RCLLER(IDA,DUMY,DUMY,MDRC(49),DUMY)                                  00000400
0051              CALL RCLLER(NBF,DUMY,DUMY,MDRC(50),DUMY)                                  00000405
0052              CALL RCLLER(IMS,4,DUMY,DUMY,MERC(51),DUMY)                                00000410
0053              CALL RCLLER(ISEC,DUMY,DUMY,MERC(52),DUMY)                                 00000415
0054              MDRC(53)=MERC(56)                                                         00000420
0055              CALL RCLLPF(MDFI(55),MERC(58),8,3)                                        00000425
0056              CALL RCLLPF(MDFI(56),MERC(56),8,3)                                        00000430
0057              CALL RCLLPF(MDFI(57),MERC(58),4,-1)                                       00000435
0058              CALL RCLLPF(MDFI(60),MERC(59),4,2)                                        00000440
0059              CALL RCLLPF(MDFI(61),MERC(60),4,2)                                        00000445
0060              MDFC(61)=MDFC(59)                                                         00000450
0061              CALL RCLLPF(MDFI(62),MERC(62),4,2)                                        00000455
0062              CALL MOVE(MDFI(58),MERC(63),2)                                            00000460
0063              MDFC(63)=MDFI(215)                                                        00000464
0064              CALL ICVI(MDFI(91),MERC(67),4)                                            00000470
0065              DO 30 I=1,15                                                              00000475
0066              JJ=215+I                                                                  00000480
0067              KK=66+2*I                                                                 00000485
0068           30 CALL RCLLPF(MDFI(JJ),MERC(KK),8,3)                                        00000490
0069              DO 35 I=1,14                                                              00000495
0070              JJ=230+I                                                                  00000500
0071              KK=121+2*I                                                                00000505
0072           35 CALL RCLLPF(MDFI(JJ),MERC(KK),8,3)                                        00000510
0073              DO 40 I=1,8                                                               00000515
0074              JJ=244+I                                                                  00000520
0075              KK=157+2*I                                                                00000525
0076           40 CALL RCLLPF(MDFI(JJ),MERC(KK),8,3)                                        00000530
                C                                                                           00000535
                C     CONVERT STATION DESIGNATION FIELD: LATITUDE, LONGITUDE, P WAVE        00000540
                C     RESIDUAL, S WAVE RESIDUAL, LG RESIDUAL, LR RESIDUAL, LG               00000545
                C     LENGTH, STATION AZIMUTH, STATION ELEVATION, AND EPICENTRAL            00000550
                C     DISTANCE                                                              00000555
                C                                                                           00000560
0077              MDRC(ICPP+99)=MERC(8)                                                     00000565
0078              MERC(56)=MDFI(6)                                                          00000570
0079              CALL RCLLPF(MDFI(92),MERC(ICPP+99),8,3)                                   00000575
0080              CALL RCLLPF(MDFI(93),MERC(ICPP+101),8,3)                                  00000580
0081              DO 55 I=1,6                                                               00000585
0082              ICOPA=71+2*I                                                              00000590
0083              IEVEN=ICOPA-1                                                             00000595
0084              IF (IMDFI(ICOPA).EQ.IMERI(27)+1) IMDFI(IEVEN)=IMDFI(IEVEN)+86400          00000600
0085              RSEC=IMDFI(IEVEN)-IMERI(26)                                               00000605
0086              IF (RSEC.LT.0.0) RSEC=RSEC+1.*SEC=4                                       00000610
0087           55 CALL RCLLPF(MDFI,MERC(ICPP+102),4,-1)                                     00000615
0088              CALL ICVR(MDFI(86),MERC(ICPP+103),4)                                      00000620
0089              CALL RCLLPF(MDFI(84),MERC(ICPP+108),8,3)                                  00000625
0090              CALL RCLLPF(MDFI(82),MERC(ICPP+110),4,-1)                                 00000630
0091              CALL RCLLPF(MDFI(65),MERC(ICPP+111),8,3)                                  00000635
0092              CALL MOVE(9,MERC,MDRC,IC,680,680,680)                                     00000640
0093              JJ=IMDFI(2)                                                               00000645
0094              DO 50 I=1,JJ                                                              00000650
0095              CALL RIREAD(16,DATA,IMDFI(15),IC,650,680,680)                             00000655
0096           50 CALL RIWRIT(16,DATA,IMDFI(15),IC,680,680,680)                             00000660
0097              CALL RIFREE(IMDFI(4),1,IC)                                                00000665
0098           60 CALL RIFREE(IMDFI(3),1,IC)                                                00000670
0099              IF (IC) 60,70,60                                                          00000675
0100           60 WRITE (6,100) IC,IC                                                       00000680
0101              STOP                                                                      00000685
0102           90 WRITE (6,110)                                                             00000690
                C     CLOSE TAPES                                                           00000695
0103              CALL RIFREE('REWIND',9,ERCTRL,80,,C)                                      00000700
0104              CALL RIFREE('REWIND',4,1,1,IC)                                            00000705
0105              CALL RIFREE('REWIND',3,1,1,IC)                                            00000710
0106              CALL RIFREE('REWIND',2,1,1,IC)                                            00000715
0107              CALL RIFREE('CLOSE',4,1,1,IC)                                             00000720
0108              CALL RIFREE('CLOSE',3,1,1,IC)                                             00000725
0109              STOP                                                                      00000730
0110          100 FORMAT(1HO,' ABNORMAL COMPLETION, ICODE = ',A4,I10)                       00000735
0111          110 FORMAT(1HO,' CONVERSION COMPLETE')                                        00000740
0112              END                                                                       00000745

                                        EQUIVALENCE DATA MAP
SYMBOL    LOCATION      SYMBOL    LOCATION      SYMBOL    LOCATION      SYMBOL    LOCATION      SYMBOL    LOCATION
MERC      00002C        IMDFI     000DFC        DATA      00C0FC        TAPENO    000100        IMDFI     004138
MDRC      CC4136
```

INITIAL

```
UNIX fortran iv v01-11  source listing          page 001

      c     this routine initializes the signal measurement data base
0001          real*4 evbuf(32), rnam1(4), rnam2(4)
0002          integer*4 ndsg, nev, intev, ndr, mndr, nfb
0003          logical*1 temp1(4), temp2(4)
0004          equivalence (evbuf(1),ndsg), (evbuf(30),nxev), (rnam1(2),temp1)
0005          equivalence (rnam2(2),temp2)
0006          data rnam1/'sigm','    ','.dat',0./,rnam2/'sigm','    ','.fre',0./
      c
0007          call setfil (1,'sigms.dir')
0008          define file 1 (32000,128,u,nxrec1)
0009          call setfil (3,'sigmsdir.fre')
0010          define file 3 (126,1024,u,nxrec3)
0011          read (1'1,end=100) nev,intev,ndr,mndr
0012          if (nev.eq.0) go to 100
      c       delete event data files and freeblock files
0014          nxev=intev
0015          do 10 k=1,nev
0016          read (1'nxev) evbuf
0017          encode (4,12,temp1) ndsg
0018       12 format (i4)
0019          decode (4,14,temp1) temp1
0020          encode (4,14,temp1) temp1
0021       14 format (4i1)
0022          call setfil (2,rnam1,64)
0023          define file 2 (32000,256,u,nxrec2)
0024          endfile 2
0025          encode (4,12,temp2) ndsg
0026          decode (4,14,temp2) temp2
0027          encode (4,14,temp2) temp2
0028          call setfil (4,rnam2,64)
0029          define file 4 (126,1024,u,nxrec4)
0030          endfile 4
0031       10 continue
0032      100 continue
      c       initialize pointers
0033          nev=0
0034          intev=0
0035          ndr=1
0036          mndr=32000
0037          write (1'1) nev,intev,ndr,mndr
0038          nfb=0
0039          write (3'1) nfb
0040          endfile 1
0041         endfile 3
0042          stop
0043          end
```

```
UNIX fortran iv v01-11  source listing          page 001

      c     this routine accesses the 9-track signal measurement tape
      c     and does the necessary conversions and reformatting before
      c     supplying the information to subroutine update
      c     all info. on a tape for an event should be consecutive
      c
0001        real*4 esinfo(141),mslab(62),stan(50),head(375),data(100)
0002        integer*4 ndsg,idstat,ievr,istr,iflgev,iflgst,lndsg,ievorg(5)
0003        integer*4 idpth
0004        logical*1 chbuf(1500),temp
0005        equivalence (esinfo(1),ndsg),(esinfo(2),ievorg),(esinfo(7),evlat),
           1       (esinfo(8),evlon), (esinfo(9),evmb), (esinfo(10),idpth),
           2       (esinfo(11),iflgev),(esinfo(12),stat),(esinfo(13),idstat),
           3       (esinfo(14),iflgst),(esinfo(15),contr),(esinfo(140),ievr),
           4     . (esinfo(141),istr)
0006        equivalence (chbuf,head)
      c
0007        data stan/'anmo','anto','boco','chto','nors','gumo',
           1          'maio','lasa','nwao','grfo','shio','tato',
           2          'snzo','ilpa','alpa','ctao','zobo','kaao',
           3          'majo','kono','bfak','ctao','chgo','tnak',
           4          'tloo','eiao','kono','ogdo','kipo','alqo',
           5          'zlpo','mato','hnme','rkon','ksrs','atak',
           6          'ucak','cnak','njak','    ','cs01','cs02',
           7          'cs03','cs04','cs05','cs06','cs07','cs08',
           8          2*'    '/,   contr/'ensc'/,  lndsg/0/
      c
0008        data nmeas/25/, mslab/'ms01','ms02','ms03','ms04','ms05',
           1        'ms06','ms07','ms08','ms09','ms10','ms11',
           2        'ms12','ms13','ms14','ms15','ms16','ms17',
           3        'ms18','ms19','ms20','ms21','ms22','ms23',
           4        'ms24','ms25'/,  blank/'    '/
0009        data eoi/'eoi0'/
      c
0010        common /arg/ esinfo
0011        common /asvar/ nxrec1,nxrec2,nxrec3,nxrec4
      c
      c     open directory and directory freeblock
0012        call setfil (1,'sigms.dir')
0013        define file 1 (32000,128,u,nxrec1)
0014        call setfil (3,'sigmsdir.fre')
0015        define file 3 (126,1024,u,nxrec3)
      c
0016      5 continue
      c     read in header
      c
0017        lunit=0
0018        lenh=1500
0019        lend=400
      c     set all state flags
0020        istate=259
0021        call intape (lunit,head,lenh,isw,istate)
0022        if (isw.ne.1) go to 995
0024        if (lenh.ne.80) go to 7
0026        do 8 k=1,4
0027        call etoa (chbuf(k),temp)
0028        chbuf(k)=temp
0029      8 continue
```

```
UNIX fortran iv v01-11  source listing          page 002

0030          if (head(1).eq.eoi) go to 995
0032        7 continue
      c       convert ebcdic to ascii
      c
0033          do 10 k=29,1500
0034          call etoa (chbuf(k),temp)
0035          chbuf(k)=temp
0036       10 continue
      c
      c
      c       decode site number and use to obtain station name
0037          decode (4,100,head) idstat
0038      100 format (66(4x),i4)
0039          stat=stan(idstat)
      c       decode designation number and check to see if it is a new event
0040          decode (4,150,head) ndsg
0041      150 format (11(4x),i4)
      c       check to see if event is the same as the last event
0042          if (ndsg.eq.lnsdg) go to 50
0044          lnsdg=ndsg
0045          iflgev=1
0046          iflgst=1
      c       decode origin time, latitude, longitude, mb, and depth
0047          decode (20,200,head) ievorg
0048      200 format (47(4x),5i4)
0049          decode (8,300,head) evlat
0050      300 format (53(4x),f8.3)
0051          decode (8,350,head) evlon
0052      350 format (55(4x),f8.3)
0053          decode (4,375,head) idpth
0054      375 format (57(4x),i4)
0055          decode (4,400,head) evmb
0056      400 format (58(4x),f4.2)
0057          go to 500
0058       50 continue
0059          iflgev=-1
      c       note: if event was not in data base prior to this run,
      c             iflgst is changed to 0 in subroutine update since
      c             all stations will be new
      c
      c       read in data and convert from ibm 360/44 to pdp 11/70 floating point
      c
0060          call intape (lunit,data,lend,isw,istate)
0061          if (isw.ne.1) go to 995
0063          do 60 k=1,nmeas
0064          call f360f (data(k),pdptm)
0065          data(k)=pdptm
0066       60 continue
      c
      c       load measurements and labels (max. of 62) into argument list
0067          do 70 k=1,nmeas
0068          ilab=14+k*2
0069          esinfo(ilab)=mslab(k)
0070          ims=ilab+1
0071          esinfo(ims)=data(k)
0072       70 continue
      c       insert blank after last measurement
```

```
UNIX fortran iv v01-11  source listing          page 003

0073          if (nmeas.ne.62) esinfo(ilab+2)=blank
      c
0075      500 continue
0076          call update (ierr)
0077          if (ierr.eq.1) go to 995
0079          go to 5
0080      995 endfile 1
0081          endfile 3
0082          stop
0083          end
```

```
UNIX fortran iv vØ1-11  source listing          page ØØ1

ØØØ1          subroutine update (ierr)
        c     this routine establishes the signal measurement data base
        c     esinfo contains pertinent event-station info and flags
        c     iflgev=1      search for event
        c     iflgev=Ø      add event
        c     iflgev=-1     update previous event
        c
        c     iflgst=1      search for station
        c     iflgst=Ø      add station
        c     iflgst=-1     update previous station
        c
        c     ierr=Ø     no errors
        c     ierr=1     error inestablishing data base
        c
        c     logical unit=1    directory file
        c                 =2    event data file
        c                 =3    freeblock file for directory
        c                 =4    freeblock file for event data file
        c
ØØØ2          real*4 esinfo(141),meas(2,62),key(4),dummy(2Ø),dumy(15)
ØØØ3          real*4 dum(9),evbuf(32),stbuf(32),msbuf(64),rnam1(4),rnam2(4)
ØØØ4          integer*4 nev,intev,ndr,mndr,ndar,mndar,ndsg,lstev,nxev,nstat
ØØØ5          integer*4 intst,idstat,ncon,lstst,nxst,ievr,istr,msrec,lstms
ØØØ6          integer*4 nxms,nfb,ifrec,iflgev,iflgst,ievorg(5),idpth
ØØØ7          logical*1 temp1(4),temp2(4)
ØØØ8          equivalence (esinfo(1),ndsg),(esinfo(2),ievorg),(esinfo(7),evlat),
             1       (esinfo(8),evlon), (esinfo(9),evmb), (esinfo(1Ø),idpth),
             2       (esinfo(11),iflgev), (esinfo(12),stat), (esinfo(13),idstat),
             3       (esinfo(14),iflgst), (esinfo(15),contr), (esinfo(16),meas),
             4       (esinfo(14Ø),ievr), (esinfo(141),istr)
ØØØ9          equivalence (evbuf(29),lstev),(evbuf(3Ø),nxev), (evbuf(31),nstat),
             1         (evbuf(32),intst), (stbuf(18),lstst), (stbuf(19),nxst),
             2         (stbuf(2Ø),ncon)
ØØ1Ø          equivalence (rnam1(2),temp1), (rnam2(2),temp2)
ØØ11          data blank/'    '/, dummy/2Ø*Ø./, dumy/15*Ø./, dum/9*Ø./
ØØ12          data rnam1/'sigm','    ','.dat',Ø./,rnam2/'sigm','    ','.fre',Ø./
ØØ13          common /arg/ esinfo
ØØ14          common /asvar/ nxrec1,nxrec2,nxrec3,nxrec4
        c
        c     construct event file and freeblock names and open files
ØØ15          encode (4,6,temp1) ndsg
ØØ16        6 format (i4)
ØØ17          decode (4,7,temp1) temp1
ØØ18          encode (4,7,temp1) temp1
ØØ19        7 format (4i1)
ØØ2Ø          encode (4,6,temp2) ndsg
ØØ21          decode (4,7,temp2) temp2
ØØ22          encode (4,7,temp2) temp2
ØØ23          call setfil (2,rnam1)
ØØ24          define file 2 (32ØØØ,256,u,nxrec2)
ØØ25          call setfil (4,rnam2)
ØØ26          define file 4 (126,1Ø24,u,nxrec4)
        c
ØØ27          ierr=Ø
ØØ28          read (1'1) nev,intev,ndr,mndr
ØØ29          if (iflgev.eq.-1) go to 15Ø
ØØ31          if (iflgev.eq.1) go to 1ØØ
```

```
UNIX fortran iv vØ1-11  source listing          page ØØ2

ØØ33       1 continue
      c       add new event to data base (at top of list)
ØØ34          if (ndr.ge.mndr) go to 995
ØØ36          if (nev.ne.Ø) read (1'intev) evbuf
ØØ38          call chkfre (3,Ø,nfb,ifrec)
      c       modify backward pointer of old event
ØØ39          ndr=ndr+1
ØØ4Ø          lstev=ndr
ØØ41          if (nfb.gt.Ø) lstev=ifrec
ØØ43          if (nev.ne.Ø) write (1'intev) evbuf
      c       reset intev and appropriate pointers in new event
ØØ45          nxev=intev
ØØ46          intev=lstev
ØØ47          lstev=Ø
ØØ48          nstat=Ø
ØØ49          intst=Ø
ØØ5Ø          write (1'intev) ndsg,ievorg,evlat,evlon,evmb,idpth,dummy,
             1             lstev,nxev,nstat,intst
ØØ51          ievr=intev
ØØ52          nev=nev+1
ØØ53          write (1'1) nev,intev,ndr,mndr
      c       initialize data file
ØØ54          ndar=1
ØØ55          mndar=32ØØØ
ØØ56          write (1'1) ndar,mndar
      c       initialize freeblock
ØØ57          nfb=Ø
ØØ58          write (4'1) nfb
      c      set station flag to add stations
ØØ59          iflgst=Ø
ØØ6Ø          go to 15Ø
ØØ61     1ØØ continue
      c       search for event
ØØ62          key(1)=ndsg
ØØ63          call search (1,key,ievr,istr,icon,msrec,msind,isrch)
ØØ64          if (isrch.eq.1) go to 1
ØØ66     15Ø continue
ØØ67          read (1'ievr) evbuf
ØØ68          if (iflgst.eq.-1) go to 25Ø
ØØ7Ø          if (iflgst.eq.1) go to 2ØØ
ØØ72     151 continue
      c       add new station to top of list
ØØ73          if (ndr.ge.mndr) go to 995
      c       check to see if there is room in the data file
ØØ75          read (2'1) ndar,mndar
ØØ76          if (ndar.ge.mndar) go to 995
ØØ78          if (nstat.ne.Ø) read (1'intst) stbuf
ØØ8Ø          call chkfre (3,Ø,nfb,ifrec)
      c       modify backward pointer of old station
ØØ81          ndr=ndr+1
ØØ82          lstst=ndr
ØØ83          if (nfb.gt.Ø) lstst=ifrec
ØØ85          if (nstat.ne.Ø) write (1'intst) stbuf
      c       reset intst and appro. pointers in new station
ØØ87          nxst=intst
ØØ88          intst=lstst
ØØ89          lstst=Ø
```

```
UNIX fortran iv vØ1-11  source listing          page ØØ3

ØØ9Ø          nstat=nstat+1
ØØ91          write (1'ievr) evbuf
      c       count measurements
ØØ92          do 1Ø k=1,62
ØØ93          if (meas(1,k).eq.blank) go to 15
ØØ95       1Ø continue
ØØ96          nmeas=k
ØØ97       15 continue
ØØ98          if (k.ne.62) nmeas=k-1
Ø1ØØ          ncon=1
Ø1Ø1          call chkfre (4,Ø,nfb,ifrec)
Ø1Ø2          ndar=ndar+1
Ø1Ø3          msrec=ndar
Ø1Ø4          if (nfb.gt.Ø) msrec=ifrec
Ø1Ø6          lstms=Ø
Ø1Ø7          nxms=Ø
Ø1Ø8          if (nmeas.lt.31) go to 2Ø
Ø11Ø          if (ndar.ge.mndar) go to 995
Ø112          call chkfre (4,Ø,nfb,ifrec)
Ø113          ndar=ndar+1
Ø114          nxms=ndar
Ø115          if (nfb.gt.Ø) nxms=ifrec
Ø117       2Ø continue
Ø118          write (2'msrec) ((meas(i,j),j=1,31),i=2,1,-1),lstms,nxms
Ø119          if (nmeas.lt.31) go to 25
Ø121          lstms=msrec
Ø122          null=Ø
Ø123          write (2'nxms) ((meas(i,j),j=32,62),i=2,1,-1),lstms,null
Ø124       25 continue
Ø125          write (1'intst) stat,idstat,dumy,lstst,nxst,ncon,contr,msrec,
             1            nmeas,dum
Ø126          write (1'1) nev,intev,ndr,mndr
Ø127          write (2'1) ndar,mndar
Ø128          endfile 2
Ø129          endfile 4
Ø13Ø          return
Ø131     2ØØ continue
      c       search for station
Ø132          key(2)=stat
Ø133          call search (2,key,ievr,istr,icon,msrec,msind,isrch)
Ø134          if (isrch.eq.1) go to 151
Ø136     25Ø continue
Ø137          read (1'istr) stbuf
      c       search for contractor
Ø138          key(3)=contr
Ø139          call search (3,key,ievr,istr,icon,msrec,msind,isrch)
Ø14Ø          if (isrch.eq.Ø) go to 3ØØ
      c add new contractor block if there is room
Ø142          if (ncon.lt.4) go to 5Ø
Ø144          write (6,3Ø)
Ø145       3Ø format('  all possible contr. blocks already full')
Ø146          go to 997
Ø147       5Ø continue
      c       count measurements
Ø148          do 6Ø k=1,62
Ø149          if (meas(1,k).eq.blank) go to 65
Ø151       6Ø continue
```

```
UNIX fortran iv v01-11  source listing          page 004

0152          nmeas=k
0153       65 continue
0154          if (k.ne.62) nmeas=k-1
0156          ncon=ncon+1
0157          read (2'1) ndar,mndar
0158          if (ndar.ge.mndar) go to 995
0160          call chkfre (4,0,nfb,ifrec)
0161          ndar=ndar+1
0162          msrec=ndar
0163          if (nfb.gt.0) msrec=ifrec
0165          lstms=0
0166          nxms=0
0167          if (nmeas.lt.31) go to 70
0169          if (ndar.ge.mndar) go to 995
0171          call chkfre (4,0,nfb,ifrec)
0172          ndar=ndar+1
0173          nxms=ndar
0174          if (nfb.gt.0) nxms=ifrec
0176       70 continue
0177          write (2'msrec) ((meas(i,j),j=1,31),i=2,1,-1),lstms,nxms
0178          if (nmeas.lt.31) go to 75
0180          lstms=msrec
0181          null=0
0182          write (2'nxms) ((meas(i,j),j=32,62),i=2,1,-1),lstms,null
0183       75 continue
      c       store contractor info
0184          ic=21+3*(ncon-1)
0185          stbuf(ic)=contr
0186          stbuf(ic+1)=msrec
0187          stbuf(ic+2)=nmeas
0188          write (1'istr) stbuf
0189          write (1'1) nev,intev,ndr,mndr
0190          write (2'1) ndar,mndar
0191          endfile 2
0192          endfile 4
0193          return
0194      300 continue
      c       loop over measurements---searching and updating
0195          do 90 k=1,62
0196          if (meas(1,k).eq.blank) go to 999
0198          key(4)=meas(1,k)
0199          call search (4,key,ievr,istr,icon,msrec,msind,isrch)
0200          if (isrch.eq.1) go to 80
0202          read (2'msrec) msbuf
0203          msbuf(msind)=meas(2,k)
0204          write (2'msrec) msbuf
0205          go to 90
0206       80 continue
      c       when measurement not found, msrec is last meas. record
0207          msind=mod(nmeas,31)+1
0208          read (2'msrec) msbuf
0209          if (msind.le.31) go to 85
0211          if (ndar.ge.mndar) go to 995
0213          msind=1
0214          call chkfre (4,0,nfb,ifrec)
0215          ndar=ndar+1
0216          nxms=ndar
```

```
UNIX fortran iv v01-11  source listing          page 005

0217          if (nfb.gt.0) nxms=ifrec
0219          msbuf(64)=nxms
0220          write (2'msrec) msbuf
0221          do 84 n=1,64
0222       84 msbuf(n)=0.0
0223          msbuf(63)=msrec
0224          msrec=nxms
0225       85 continue
0226          msbuf(msind)=meas(2,k)
0227          msbuf(msind+31)=meas(1,k)
0228          write (2'msrec) msbuf
0229          nmeas=nmeas+1
0230          stbuf(icon+2)=nmeas
0231          write (1'istr) stbuf
0232       90 continue
0233          go to 999
0234      995 continue
0235          write (6,996)
0236      996 format ('  files are full')
0237      997 ierr=1
0238      999 endfile 2
0239          endfile 4
0240          return
0241          end
```

```
UNIX fortran iv vØ1-11  source listing          page ØØ1

ØØØ1          subroutine search (iflag,key,evrec,strec,icon,msrec,msind,isrch)
       c      this routine searches for a specific keyword and returns the
       c      appropriate record and/or index
       c
       c      iflag=Ø     full search (needs all keys)
       c      iflag=1     search for event
       c      iflag=2     search for station (needs evrec)
       c      iflag=3     search for contractor (needs strec)
       c      iflag=4     search for measurement (needs icon)
       c         (returns rec # in 'sigmxxxx.dat' and index into rec)
       c      key(1)=event designation
       c      key(2)=station name
       c      key(3)=contractor name
       c      key(4)=measurement mnemonic
       c      isrch=Ø     successful search
       c      isrch=1     unsuccessful search
       c
ØØØ2          integer*4 nev,intev,ndr,mndr,evrec,strec,msrec,nmeas
ØØØ3            integer*4 ndsg,nxev,nstat,intst,nxst,nxms,icontr(12)
ØØØ4          real*4 key(4),buff(32),msbuff(64)
ØØØ5          equivalence (buff(1),ndsg,stat), (buff(19),nxst), (buff(3Ø),nxev)
ØØØ6          equivalence (buff(31),nstat), (buff(32),intst), (msbuff(64),nxms),
             1     (buff(21),icontr)
ØØØ7          common /asvar/ nxrec1,nxrec2,nxrec3,nxrec4
ØØØ8          isrch=Ø
ØØØ9          if (iflag.eq.Ø) go to 1ØØ
ØØ11          go to (1ØØ,2ØØ,3ØØ,4ØØ), iflag
ØØ12       1ØØ read (1'1) nev,intev,ndr,mndr
ØØ13          if (nev.eq.Ø) go to 995
       c      search through events
ØØ15          nxev=intev
ØØ16          do 1Ø k=1,nev
ØØ17          evrec=nxev
ØØ18          read (1'nxev) buff
ØØ19          if (key(1).eq.float(ndsg)) go to 15
ØØ21       1Ø continue
ØØ22          write (6,14) key(1)
ØØ23       14 format('  event: ',f4.Ø,' not found')
ØØ24          isrch=1
ØØ25          go to 999
ØØ26       15 continue
ØØ27          if (iflag.eq.Ø) go to 2Ø1
ØØ29          return
ØØ3Ø       2ØØ continue
       c      search through stations given base vector (event) record #
ØØ31          if (evrec.eq.Ø) go to 995
ØØ33          read (1'evrec) buff
ØØ34       2Ø1 if (nstat.eq.Ø) go to 995
ØØ36          kount=nstat
ØØ37          nxst=intst
ØØ38          do 2Ø k=1,kount
ØØ39          strec=nxst
ØØ4Ø          read (1'nxst) buff
ØØ41          if (key(2).eq.stat) go to 25
ØØ43       2Ø continue
ØØ44          write (6,24) key(2)
ØØ45       24 format('  station: ',a4,' not found')
```

```
UNIX fortran iv v01-11  source listing          page 002

0046          isrch=1
0047          go to 999
0048       25 continue
0049          if (iflag.eq.0) go to 301
0051          return
0052      300 continue
      c       search station record for contractor index
0053          if (strec.eq.0) go to 995
0055          read (1'strec) buff
0056      301 continue
0057          do 30 icon=21,30,3
0058          if (key(3).eq.buff(icon)) go to 35
0060       30 continue
0061          write (6,34) key(3)
0062       34 format('  contractor: ',a4,' not found')
0063          isrch=1
0064          go to 999
0065       35 continue
0066          if (iflag.eq.0) go to 400
0068          return
0069      400 continue
0070          ioff=icon-20
0071          msrec=icontr(ioff+1)
0072          nmeas=icontr(ioff+2)
0073          if (nmeas.eq.0) go to 995
0075          nblk=nmeas/31+1
0076          if (mod(nmeas,31).eq.0) nblk=nblk-1
0078          nxms=msrec
0079          do 45 n=1,nblk
      c       set msrec to current rec #
0080          msrec=nxms
0081          read (2'nxms) msbuff
      c     search through measurement mnemonics
0082          do 40 k=32,62
0083          if (key(4).eq.msbuff(k)) go to 50
0085       40 continue
0086       45 continue
0087          write (6,46) key(4)
0088       46 format('  measurement: ',a4,' not found')
0089          isrch=1
0090          go to 999
0091       50 continue
0092          msind=k-31
0093          go to 999
0094      995 write (6,996)
0095      996 format('  error on search')
0096          stop
0097      999 return
0098          end
```

```
UNIX fortran iv v01-11  source listing          page 001

0001          subroutine chkfre (lunit,iflag,nfb,ifrec)
      c       this subroutine checks the appropriate freeblock file and
      c       returns the total of free blocks before this call and
      c       the next free record
      c       iflag=0     returns free record in ifrec
      c       iflag=1     freeblock extended with ifrec
      c
0002          integer*4 nfb,nnfb,ifrec,freb(256)
0003          common /asvar/ nxrec1,nxrec2,nxrec3,nxrec4
0004          read(lunit'1) nfb
0005          if (nfb.eq.0.and.iflag.eq.0) return
0007          irecfb=nfb/256+2
0008          if (mod(nfb,256).eq.0) irecfb=irecfb-1
0010          read (lunit'irecfb) freb
0011          if (iflag.eq.1) go to 10
0013          ind=mod(nfb,256)
0014          if (ind.eq.0) ind=256
0016          ifrec=freb(ind)
0017          nnfb=nfb-1
0018          go to 20
0019       10 continue
      c       find next freeblock entry
0020          ind=mod(nfb,256)+1
0021          if (ind.eq.1) irecfb=irecfb+1
0023          freb(ind)=ifrec
0024          nnfb=nfb+1
0025          write (lunit'irecfb) freb
0026       20 continue
      c        modify freeblock header
027           write (lunit'1) nnfb
0028          return
0029          end
```

NAVE
(PAGE 1 OF 11)

```
UNIX fortran iv v01-11  source listing          page 001

      c     this routine performs a network average for the first
      c     'ndisc' discriminants for each station for a given event
      c
0001        real*4 stat,contr,mslab,rmeas(100),rlab(100),evbuf(32),stbuf(32)
0002        real*4 key(4), rmx(650), discr(100)
0003        integer*4 ndsg,nret,ievr,intst,nxst,istr,msrec,nstat
      c     equivalence (evbuf(32),intst),(evbuf(31),nstat), (stbuf(19),nxst),
      c    1           (stbuf(1),stat)
0004        data contr/'ensc'/, rmx/650*0./
      c
0005        common /flags/ iflgms,iflgev
0006        common /asvar/ nxrec1,nxrec2,nxrec3,nxrec4
      c
0007        call setfil (1,'sigms.dir')
0008        define file 1 (32000,128,u,nxrec1)
      c     # of discriminants to average
0009        ndisc=21
0010        write (6,100)
0011    100 format (' input event designation #, (i4)')
0012        read (5,150) ndsg
0013    150 format (i4)
0014        key(1)=ndsg
0015        call search (1,key,ievr,istr,icon,msrec,msind,isrch)
0016        if (isrch.eq.1) go to 999
0018        read (1'ievr) evbuf
0019        iflgms=0
0020        iflgev=1
0021        nxst=intst
0022        do 10 k=1,nstat
0023        read (1'nxst) stbuf
0024        call access (ndsg,stat,contr,mslab,nret,rmeas,rlab,ievr)
      c     load matrix with measurements
0025        do 20 j=1,ndisc
0026        ind=j+(k-1)*ndisc
0027        rmx(ind)=rmeas(j)
0028     20 continue
0029     10 continue
      c
      c     convert from integer*4 to integer*2
0030        nsta=nstat*1
0031        call mbias (rmx,nsta,ndisc,discr)
0032        write (6,200) (rlab(i),discr(i),i=1,ndisc)
0033    200 format (20(5(a4,' : ',f7.3,5x)/))
0034    999 endfile 1
0035        stop
0036        end
```

```
UNIX fortran iv v01-11  source listing          page 001

0001          subroutine search (iflag,key,evrec,strec,icon,msrec,msind,isrch)
      c       this routine searches for a specific keyword and returns the
      c       appropriate record and/or index
      c
      c       iflag=0     full search (needs all keys)
      c       iflag=1     search for event
      c       iflag=2     search for station (needs evrec)
      c       iflag=3     search for contractor (needs strec)
      c       iflag=4     search for measurement (needs icon)
      c          (returns rec # in 'sigmxxxx.dat' and index into rec)
      c       key(1)=event designation
      c       key(2)=station name
      c       key(3)=contractor name
      c       key(4)=measurement mnemonic
      c       isrch=0     successful search
      c       isrch=1     unsuccessful search
      c
0002          integer*4 nev,intev,ndr,mndr,evrec,strec,msrec,nmeas
0003            integer*4 ndsg,nxev,nstat,intst,nxst,nxms,icontr(12)
0004          real*4 key(4),buff(32),msbuff(64)
0005          equivalence (buff(1),ndsg,stat), (buff(19),nxst), (buff(30),nxev)
0006          equivalence (buff(31),nstat), (buff(32),intst), (msbuff(64),nxms),
             1    (buff(21),icontr)
0007          common /asvar/ nxrec1,nxrec2,nxrec3,nxrec4
0008          isrch=0
0009          if (iflag.eq.0) go to 100
0011          go to (100,200,300,400), iflag
0012     100  read (1'1) nev,intev,ndr,mndr
0013          if (nev.eq.0) go to 995
      c       search through events
0015          nxev=intev
0016          do 10 k=1,nev
0017          evrec=nxev
0018          read (1'nxev) buff
0019          if (key(1).eq.float(ndsg)) go to 15
0021       10 continue
0022          write (6,14) key(1)
0023       14 format('  event: ',f4.0,' not found')
0024          isrch=1
0025          go to 999
0026       15 continue
0027          if (iflag.eq.0) go to 201
0029          return
0030     200  continue
      c     search through stations given base vector (event) record #
0031          if (evrec.eq.0) go to 995
0033          read (1'evrec) buff
0034     201  if (nstat.eq.0) go to 995
0036          kount=nstat
0037          nxst=intst
0038          do 20 k=1,kount
0039          strec=nxst
0040          read (1'nxst) buff
0041          if (key(2).eq.stat) go to 25
0043       20 continue
0044          write (6,24) key(2)
0045       24 format('  station: ',a4,' not found')
```

```
UNIX fortran iv v01-11  source listing          page 002

0046          isrch=1
0047          go to 999
0048       25 continue
0049          if (iflag.eq.0) go to 301
0051          return
0052      300 continue
      c       search station record for contractor index
0053          if (strec.eq.0) go to 995
0055          read (1'strec) buff
0056      301 continue
0057          do 30 icon=21,30,3
0058          if (key(3).eq.buff(icon)) go to 35
0060       30 continue
0061          write (6,34) key(3)
0062       34 format('  contractor: ',a4,' not found')
0063          isrch=1
0064          go to 999
0065       35 continue
0066          if (iflag.eq.0) go to 400
0068          return
0069      400 continue
0070          ioff=icon-20
0071          msrec=icontr(ioff+1)
0072          nmeas=icontr(ioff+2)
0073          if (nmeas.eq.0) go to 995
0075          nblk=nmeas/31+1
0076          if (mod(nmeas,31).eq.0) nblk=nblk-1
0078          nxms=msrec
0079          do 45 n=1,nblk
      c       set msrec to current rec #
0080          msrec=nxms
0081          read (2'nxms) msbuff
      c     search through measurement mnemonics
0032          do 40 k=32,62
0033          if (key(4).eq.msbuff(k)) go to 50
0085       40 continue
0086       45 continue
0087          write (6,46) key(4)
0088       46 format('  measurement: ',a4,' not found')
0089          isrch=1
0090          go to 999
0091       50 continue
0092          msind=k-31
0093          go to 999
0094      995 write (6,996)
0095      996 format('  error on search')
0096          stop
0097      999 return
0098          end
```

```
UNIX fortran iv vØ1-11  source listing          page ØØ1

ØØØ1          subroutine access (ndsg,stat,contr,mslab,nret,rmeas,rlab,ievr)
       c      this routine accesses the signal measurement data base
       c           ndsg=event designation #
       c           stat=station name (if = '****',given measurement is returned for all
       c                     stations for event)
       c           contr=contractor name
       c           iflgms=Ø  return all measurements for station (in rmeas) with assoc.
       c                     labels (in rlab)--------warning: it should be noted that
       c                     there is no guarantee all the meas. will be in the same
       c                     order across stations
       c           iflgms=1  return only indicated measurement
       c           mslab=label of desired measurement
       c           nret=# of measurements returned
       c           rmeas=measurement values
       c           rlab=associated measurement labels
       c           iflgev=Ø   search for event
       c           iflgev=1   use previous or supplied event record # (ievr)
       c           ievr=event record #
       c
ØØØ2          real*4 stat,contr,mslab,rmeas(1),rlab(1),evbuf(32),stbuf(32)
ØØØ3          real*4 buff(64),key(4),rnam1(4)
ØØØ4          integer*4 ndsg,ievr,istr,msrec,nxms,nstat,intst,nxst,icontr(12)
ØØØ5          integer*4 nret
ØØØ6          logical*1 temp1(4)
ØØØ7          equivalence (evbuf(31),nstat),(evbuf(32),intst), (stbuf(19),nxst),
              1           (buff(64),nxms), (stbuf(21),icontr)
ØØØ8          equivalence (rnam1(2),temp1)
ØØØ9          data rnam1/'sigm','    ','.dat',Ø./, star/'****'/
ØØ1Ø          common /flags/ iflgms,iflgev
ØØ11          common /asvar/ nxrec1,nxrec2,nxrec3,nxrec4
       c
       c      construct data file name and open file (directory file should already be
       c            open)
ØØ12          encode (4,1,temp1) ndsg
ØØ13        1 format (i4)
ØØ14          decode (4,2,temp1) temp1
ØØ15          encode (4,2,temp1) temp1
ØØ16        2 format (4i1)
ØØ17          call setfil (2,rnam1)
ØØ18          define file 2 (1ØØ,256,u,nxrec2)
       c
ØØ19          nret=Ø
ØØ2Ø          key(1)=ndsg
ØØ21          key(3)=contr
ØØ22          if (iflgev.eq.Ø) call search (1,key,ievr,istr,icon,msrec,msind,
              1                             isrch)
ØØ24          if (iflgev.eq.Ø.and.isrch.eq.1) go to 995
ØØ26          if (stat.eq.star) go to 5ØØ
       c      search for specified station
ØØ28          key(2)=stat
ØØ29          call search(2,key,ievr,istr,icon,msrec,msind,isrch)
ØØ3Ø          if (isrch.eq.1) go to 995
       c      search for contractor index
ØØ32          call search (3,key,ievr,istr,icon,msrec,msind,isrch)
ØØ33          if (isrch.eq.1) go to 995
ØØ35          if (iflgms.eq.1) go to 3Ø
       c      return all measurements for station
```

```
UNIX fortran iv v01-11  source listing          page 002

0037          read (1'istr) stbuf
0038          ioff=icon-20
0039          msrec=icontr(ioff+1)
0040          nret=icontr(ioff+2)
0041          nblk=nret/31+1
0042          if (mod(nret,31).eq.0) nblk=nblk-1
0044          nxms=msrec
0045          do 10 n=1,nblk
0046          read (2'nxms) buff
0047          lim=nret-(n-1)*31
0048          if (lim.gt.31) lim=31
0050          do 20 k=1,lim
0051          ist=(n-1)*31+k
0052          rmeas(ist)=buff(k)
0053          rlab(ist)=buff(k+31)
0054       20 continue
0055       10 continue
0056          go to 995
0057       30 continue
0058          key(4)=mslab
0059          call search (4,key,ievr,istr,icon,msrec,msind,isrch)
0060          if (isrch.eq.1) go to 995
0062          read (2'msrec) buff
0063          nret=1
0064          rmeas(nret)=buff(msind)
0065          go to 995
0066      500 continue
      c       return one measurement for all stations across event
0067          key(4)=mslab
0068          read (1'ievr) evbuf
0069          nxst=intst
0070          do 40 k=1,nstat
0071          istr=nxst
0072          read (1'nxst) stbuf
      c       search for contractor index
0073          call search (3,key,ievr,istr,icon,msrec,msind,isrch)
0074          if (isrch.eq.1) go to 40
      c       search for measurement
0076          call search (4,key,ievr,istr,icon,msrec,msind,isrch)
0077          if (isrch.eq.1) go to 40
0079          read (2'msrec) buff
0030          nret=nret+1
0081          rmeas(nret)=buff(msind)
0082       40 continue
0083      995 endfile 2
0084          return
0085          end
```

```
UNIX fortran iv v01-11  source listing          page 001

0001          subroutine mbias (rmx, nsta, ndisc, discr)
      c
      c       rmx - vector length >= nsta*ndisc
      c            contains the discriminants (measured)
      c       nsta - number of stations (must be <= 50)
      c       ndisc - number of discriminants
      c       discr - unbiased discriminant output
      c               vector length >= ndisc
      c
      c       this routine performs ringdahl's maximum likelihood
      c       estimating technique for network averaging
      c
      c
0002          real      rmx(1),          discr(1)
0003          common / svsto / nstat,       ndet(50),
     1                         xmag(50),    sdnois(50)
      c
      c       check number of stations
      c
0004               if (nsta.le.50) go to 2
0006               write (6,3)
0007        3      format (' maximum # of stations (50) exceeded')
0008               go to 99
0009        2      continue
0010               nstat=nsta
0011               xbias = 1000.0
0012               sigmin = 0.2
0013               sigmax = 1.0
0014               do 5 i = 1, 50
0015               sdnois(i) = 0.1
0016        5      continue
0017               do 10 i = 1, ndisc
      c
      c       initialize signal average
      c
0018               nn = 0
0019               ns = 0
0020               nave = 0
0021               sigmag = 0.0
      c
0022               do 20 j = 1, nsta
      c
      c       get discriminant measurement
      c
0023               xmag(j) = rmx(i+(j-1)*ndisc)
0024               if ( xmag(j) ) 30, 40, 50
      c
      c       noise measurement---get rid of bias
      c
0025     30        continue
0026               nn = nn + 1
0027               xmag(j) = xmag(j) + xbias
0028               sigmag = sigmag + xmag(j)
0029               nave = nave + 1
0030               ndet(j) = 0
0031               go to 20
      c
```

```
UNIX fortran iv v01-11  source listing          page 002

      c     ignore station
      c
0032     40      continue
0033             ndet(j) = -1
0034             go to 20
      c
      c     signal measurement---get rid of bias
      c
0035     50      continue
0036             ns = ns + 1
0037             xmag(j) = xmag(j) - xbias
0038             sigmag = sigmag + xmag(j)
0039             nave = nave + 1
0040             ndet(j) = 1
0041     20      continue
      c
      c     starting estimate is average signal measurement
      c
0042             if ( nn .eq. 0 .or. ns .ne. 0 ) go to 60
0044             smax = -1000.0
0045             do 70 j = 1, nsta
0046             if ( ndet(j) .ne. 0 .or. smax .gt. xmag(j) ) go to 70
0048             smax = xmag(j)
0049             indx = j
0050     70      continue
0051             ndet(indx) = 1
0052             ns = 1
0053     60      continue
0054             if ( ns .ne. 0 ) go to 80
0056             sigmag = 0.0
0057             go to 90
0058     80      continue
0059             sdsig = 0.35
0060             sigmag = sigmag/float(nave)
0061             call max2d (sigmag, sdsig, sigmin, sigmax)
      c
      c     store unbiased discriminant into output array
      c
0062     90      continue
0063             discr(i) = sigmag
0064     10      continue
      c
0065     99      continue
      c
0066             return
      c
0067             end
```

```
UNIX fortran iv vØ1-11  source listing          page ØØ1

ØØØ1        subroutine max2d (x, y, sigmin, sigmax)
      c
ØØØ2        real      likeli
      c
ØØØ3        real      z(5)
      c
ØØØ4            niter = Ø
ØØØ5            nloop = Ø
ØØØ6            deltax = Ø.ØØ9
ØØØ7            deltay = Ø.ØØ9
      c
ØØØ8      1Ø    continue
ØØØ9            z(1) = likeli (x, y)
ØØ1Ø            z(2) = likeli (x+deltax, y)
ØØ11            z(3) = likeli (x-deltax, y)
ØØ12            yhi = y + deltay
ØØ13            if ( yhi .gt. sigmax ) yhi = sigmax
ØØ15            z(4) = likeli ( x, yhi)
ØØ16            if ( ylo .lt. sigmin ) ylo = sigmin
ØØ18            ylo = y - deltay
ØØ19            z(5) = likeli (x, ylo)
ØØ2Ø            niter = niter + 1
ØØ21            idir = 1
ØØ22            zmax = z(1)
ØØ23            do 2Ø i = 2, 5
ØØ24            if ( z(i) .lt. zmax ) go to 2Ø
ØØ26            zmax = z(i)
ØØ27            idir = i
ØØ28      2Ø    continue
      c
ØØ29            if ( idir .eq. 1 ) go to 3Ø
ØØ31            if ( idir .eq. 2 ) x = x + deltax
ØØ33            if ( idir .eq. 3 ) x = x - deltax
ØØ35            if ( idir .eq. 4 ) y = y + deltay
ØØ37            if ( idir .eq. 5 ) y = y - deltay
ØØ39            if ( y .gt. sigmax ) y = sigmax
ØØ41            if ( y .lt. sigmin ) y = sigmin
ØØ43            if ( niter .gt. 1ØØ ) return
ØØ45            go to 1Ø
      c
ØØ46      3Ø    continue
ØØ47            deltax = deltax/2.Ø
ØØ48            deltay = deltay/2.Ø
ØØ49            nloop = nloop + 1
ØØ5Ø            if ( nloop .gt. 5 .or. niter .gt. 1ØØ ) return
ØØ52            go to 1Ø
      c
ØØ53        end
```

```
UNIX fortran iv v01-11  source listing          page 001

0001          real function likeli (mean, sigma)
      c
      c       this function returns the value of the
      c       log-likelihood function which is to be
      c       maximized w.r.t. the mean and  the
      c       standard deviation
      c
0002          real                mean
      c
0003          double precision    ans,          gauss,
             1                    x
      c
0004          common / svsto /    nsta,              ndet(50),
             1                    xmag(50),          sdnois(50)
      c
0005               x = 0.0d0
0006               do 10 i = 1, nsta
0007               if ( ndet(i) ) 10, 20, 30
      c
      c       detecting station
      c
0008       30      continue
0009               arg = (xmag(i) - mean)/sigma
0010               ans = exp (-arg*arg/2.0)/sigma
0011               go to 40
      c
      c       non-detecting station
      c
0012       20      continue
0013               arg = (xmag(i) - mean)/sqrt(sigma*sigma + sdnois(i)*sdnois(i))
0014               ans = gauss ( arg )
      c
0015       40      continue
0016               if ( ans .lt. 1.0d-38 ) ans = 1.0d-38
0018               x = x + dlog10 (ans)
0019       10      continue
      c
0020               likeli = x
0021               return
      c
0022          end
```

```
UNIX fortran iv vØ1-11  source listing          page ØØ1

ØØØ1          double precision function gauss (x)
      c
ØØØ2          double precision  root2,          derfc,
             1                  z
      c
ØØØ3          data              root2 / 1.414213562dØ /
      c
ØØØ4               z = -x/root2
ØØØ5               gauss = Ø.5dØ*derfc (z)
      c
ØØØ6               return
      c
ØØØ7          end
```

```
UNIX fortran iv v01-11  source listing          page 001

0001          function derfc (x)
0002          implicit double precision (a-h, o-z)
0003              derfc = 2.0d0
0004              if ( x .le. -10.0d0 ) return
0006              derfc = 0.0d0
0007              if ( x .ge. 20.0d0 ) return
0009              derfc = 1.0d0
0010              if ( x .eq. 0.0d0 ) return
0012              rootpi = 1.7724538509d0
0013              xsqr = x * x
0014              sum = 0.0d0
0015              xnum = 1.0d0
0016              term = 1.0d0
0017              if ( x .lt. 3.1d0 ) go to 10
0019              factor = dexp (-xsqr)/(rootpi*x)
0020              xsqr = 2.0d0 * xsqr
0021      20      continue
0022              sum = sum + term
0023              xnum = xnum - 2.0d0
0024              term1 = xnum*term/xsqr
0025              if ( term1/term .lt. -1.0d0 .or.
                1         dabs (term1) .lt. 1.0d-20 ) go to 30
0027              term = term1
0028              go to 20
0029      30      continue
0030              derfc = factor*sum
0031              return
0032      10      continue
0033              factor = dexp (-xsqr)*x*2.0d0/rootpi
0034              xsqr = 2.0d0*xsqr
0035      40      continue
0036              sum = sum + term
0037              if ( term/sum .lt. 1.0d-20 ) go to 50
0039              xnum = xnum + 2.0d0
0040              term = term*xsqr/xnum
0041              go to 40
0042      50      continue
0043              derfc = 1.0d0 - factor*sum
0044              return
0045          end
```

```
UNIX fortran iv v01-11  source listing          page 001

      c     this routine performs a network average for the first
      c     'ndisc' discriminants for each station for a given event
      c
0001        real*4 stat,contr,mslab,rmeas(25),rlab(25),evbuf(32),stbuf(32)
0002        real*4 key(4), rmx(150), discr(25), disc(25)
0003        integer setfil
0004        integer*4 ndsg,nret,ievr,intst,nxst,istr,msrec,nstat
0005        equivalence (evbuf(32),intst),(evbuf(31),nstat), (stbuf(19),nxst),
           1          (stbuf(1),stat)
0006        data contr/'ensc'/, rmx/150*0./
      c
0007        common /flags/ iflgms,iflgev
0008        common /asvar/ nxrec1,nxrec2,nxrec3,nxrec4
      c
0009        call setfil (1,'sigms.dir')
0010        define file 1 (100,128,u,nxrec1)
      c     # of discriminants to average
0011        ndisc=21
0012        write (6,100)
0013    100 format (' input event designation #, (i4)')
0014        read (5,150) ndsg
0015    150 format (i4)
0016        key(1)=ndsg
0017        call search (1,key,ievr,istr,icon,msrec,msind,isrch)
0018        if (isrch.eq.1) go to 999
0020        read (1'ievr) evbuf
0021        iflgms=0
0022        iflgev=1
      c     convert integer*4 to integer*2
0023        nsta=nstat*1
0024        nd=3
0025        ndsav=nd
0026        nloop=ndisc/nd+1
0027        if (mod(ndisc,nd).eq.0) nloop=nloop-1
0029        do 15 n=1,nloop
0030        nxst=intst
0031        do 10 k=1,nstat
0032        read (1'nxst) stbuf
0033        call access (ndsg,stat,contr,mslab,nret,rmeas,rlab,ievr)
      c     load matrix with measurements
0034        if (n.eq.nloop.and.mod(ndisc,nd).ne.0) nd=mod(ndisc,nd)
0036        do 20 j=1,nd
0037        ind=j+(k-1)*nd
0038        ioff=j+(n-1)*ndsav
0039        rmx(ind)=rmeas(ioff)
0040     20 continue
0041     10 continue
0042        call mbias (rmx,nsta,nd,discr)
0043        do 14 i=1,nd
0044        ind=i+(n-1)*ndsav
0045        disc(ind)=discr(i)
0046     14 continue
0047     15 continue
      c
0048        write (6,200) (rlab(i),disc(i),i=1,ndisc)
0049    200 format (20(5(a4,' : ',f7.3,5x)/))
0050    999 endfile 1
0051        stop
052        end
```

EVLIST
(PAGE 1 OF 6)

```
UNIX fortran iv v01-11  source listing          page 001

0001          real*4 evlat,evlon,evmb,stat,contr,rmeas(100),rlab(100)
0002          real*4 mslab,evbuf(32),stbuf(32)
0003          integer*4 nev,intev,ndsg,ievorg(5),idpth,intst,nstat
0004          integer*4 nxev,nxst,ncon,icontr(12),ievr,nret,idstat
0005          equivalence (evbuf(1),ndsg), (evbuf(2),ievorg), (evbuf(7),evlat),
             1        (evbuf(8),evlon), (evbuf(9),evmb), (evbuf(10),idpth),
             2        (evbuf(30),nxev), (evbuf(31),nstat), (evbuf(32),intst)
0006          equivalence (stbuf(1),stat), (stbuf(2),idstat), (stbuf(19),nxst),
             1        (stbuf(20),ncon), (stbuf(21),icontr)
0007          data contr/'ensc'/
0008          common /flags/ iflgms,iflgev
0009          common /asvar/ nxrec1,nxrec2,nxrec3,nxrec4
      c
0010          call setfil (1,'sigms.dir')
0011          define file 1 (32000,128,u,nxrec1)
      c
0012          read (1'1) nev,intev
0013          if (nev.eq.0) go to 999
      c
0015          write (6,100)
0016      100 format (' input 0 to list event info; 1 to include stat. meas.',
             1     ' (i1)')
0017          read (5,150) iflg
0018      150 format (i1)
0019        1 continue
0020          write (6,200)
0021      200 format (' input first and last event sequence #"s (2i3)',
             1     ' [<cr> to exit]')
0022          read (5,250) ifirst,ilast
0023      250 format (2i3)
0024          if (ifirst .le. 0) go to 999
0026          if (ifirst .gt. nev) go to 999
0028          if (ilast .gt. nev) ilast=nev*1
      c       loop over events
0030          nxev=intev
0031          do 10 k=1,ilast
0032          write (6,260)
0033      260 format (///)
0034          ievr=nxev
0035          read (1'nxev) evbuf
0036          if (k.lt.ifirst) go to 10
0038          write (6,300) k,ndsg,ievorg,evlat,evlon,evmb,idpth,nstat
0039      300 format (' event seq # = ',i3,5x,'event designation # = ',i4/
             1     ' origin time : ',i2,i3,2x,3(i2,1x),4x,'latitude (+n) : ',
             2     f8.3,5x,'longitude (+e) : ',f8.3/' mb = ',f4.2,5x,
             3     'depth (km) = ',i4,5x,'# of stations = ',i3)
      c       loop over stations
0040          nxst=intst
0041          do 20 j=1,nstat
0042          write (6,320)
0043      320 format(/)
0044          read (1'nxst) stbuf
0045          write (6,350) idstat,stat,(icontr(1+(i-1)*3),i=1,ncon)
0046      350 format (' station # : ',i2,10x,'station name : ',a4/
             1     ' contractors present : ',4(a4,3x))
0047          if (iflg.eq.0) go to 20
0049          iflgev=1
```

```
UNIX fortran iv vØ1-11  source listing          page ØØ2

ØØ5Ø          iflgms=Ø
ØØ51          call access (ndsg,stat,contr,mslab,nret,rmeas,rlab,ievr)
ØØ52          write (6,4ØØ) contr
ØØ53      4ØØ format (' measurements for contractor : ',a4)
ØØ54          write (6,45Ø) (rlab(i),rmeas(i),i=1,nret)
ØØ55      45Ø format (2Ø(5(a4,' : ',f9.3,5x)/))
ØØ56       2Ø continue
ØØ57       1Ø continue
ØØ58          go to 1
ØØ59      999 endfile 1
ØØ6Ø          stop
ØØ61          end
```

```
UNIX fortran iv v01-11  source listing          page 001

0001          subroutine search (iflag,key,evrec,strec,icon,msrec,msind,isrch)
      c       this routine searches for a specific keyword and returns the
      c       appropriate record and/or index
      c
      c       iflag=0     full search (needs all keys)
      c       iflag=1     search for event
      c       iflag=2     search for station (needs evrec)
      c       iflag=3     search for contractor (needs strec)
      c       iflag=4     search for measurement (needs icon)
      c          (returns rec # in 'sigmxxxx.dat' and index into rec)
      c       key(1)=event designation
      c       key(2)=station name
      c       key(3)=contractor name
      c       key(4)=measurement mnemonic
      c       isrch=0     successful search
      c       isrch=1     unsuccessful search
      c
0002          integer*4 nev,intev,ndr,mndr,evrec,strec,msrec,nmeas
0003            integer*4 ndsg,nxev,nstat,intst,nxst,nxms,icontr(12)
0004          real*4 key(4),buff(32),msbuff(64)
0005          equivalence (buff(1),ndsg,stat), (buff(19),nxst), (buff(30),nxev)
0006          equivalence (buff(31),nstat), (buff(32),intst), (msbuff(64),nxms),
             1    (buff(21),icontr)
0007          common /asvar/ nxrec1,nxrec2,nxrec3,nxrec4
0008          isrch=0
0009          if (iflag.eq.0) go to 100
0011          go to (100,200,300,400), iflag
0012      100 read (1'1) nev,intev,ndr,mndr
0013          if (nev.eq.0) go to 995
      c       search through events
0015          nxev=intev
0016          do 10 k=1,nev
0017          evrec=nxev
0018          read (1'nxev) buff
0019          if (key(1).eq.float(ndsg)) go to 15
0021       10 continue
0022          write (6,14) key(1)
0023       14 format('  event: ',f4.0,' not found')
0024          isrch=1
0025          go to 999
0026       15 continue
0027          if (iflag.eq.0) go to 201
0029          return
0030      200 continue
      c     search through stations given base vector (event) record #
0031          if (evrec.eq.0) go to 995
0033          read (1'evrec) buff
0034      201 if (nstat.eq.0) go to 995
0036          kount=nstat
0037          nxst=intst
0038          do 20 k=1,kount
0039          strec=nxst
0040          read (1'nxst) buff
0041          if (key(2).eq.stat) go to 25
0043       20 continue
0044          write (6,24) key(2)
0045       24 format('  station: ',a4,' not found')
```

```
UNIX fortran iv v01-11  source listing          page 002

0046          isrch=1
0047          go to 999
0048       25 continue
0049          if (iflag.eq.0) go to 301
0051          return
0052      300 continue
      c       search station record for contractor index
0053          if (strec.eq.0) go to 995
0055          read (1'strec) buff
0056      301 continue
0057          do 30 icon=21,30,3
0058          if (key(3).eq.buff(icon)) go to 35
0060       30 continue
0061          write (6,34) key(3)
0062       34 format('  contractor: ',a4,' not found')
0063          isrch=1
0064          go to 999
0065       35 continue
0066          if (iflag.eq.0) go to 400
0068          return
0069      400 continue
0070          ioff=icon-20
0071          msrec=icontr(ioff+1)
0072          nmeas=icontr(ioff+2)
0073          if (nmeas.eq.0) go to 995
0075          nblk=nmeas/31+1
0076          if (mod(nmeas,31).eq.0) nblk=nblk-1
0078          nxms=msrec
0079          do 45 n=1,nblk
      c       set msrec to current rec #
0080          msrec=nxms
0081          read (2'nxms) msbuff
      c     search through measurement mnemonics
0082          do 40 k=32,62
0083          if (key(4).eq.msbuff(k)) go to 50
0085       40 continue
0086       45 continue
0087          write (6,46) key(4)
0088       46 format('  measurement: ',a4,' not found')
0089          isrch=1
0090          go to 999
0091       50 continue
0092          msind=k-31
0093          go to 999
0094      995 write (6,996)
0095      996 format('  error on search')
0096          stop
0097      999 return
0098          end
```

```
UNIX fortran iv v01-11  source listing          page 001

0001          subroutine access (ndsg,stat,contr,mslab,nret,rmeas,rlab,ievr)
      c       this routine accesses the signal measurement data base
      c            ndsg=event designation #
      c            stat=station name (if = '****',given measurement is returned for all
      c                 stations for event)
      c            contr=contractor name
      c            iflgms=0  return all measurements for station (in rmeas) with assoc.
      c                      labels (in rlab)--------warning: it should be noted that
      c                      there is no guarantee all the meas. will be in the same
      c                      order across stations
      c            iflgms=1  return only indicated measurement
      c            mslab=label of desired measurement
      c            nret=# of measurements returned
      c            rmeas=measurement values
      c            rlab=associated measurement labels
      c            iflgev=0   search for event
      c            iflgev=1   use previous or supplied event record # (ievr)
      c            ievr=event record #
      c
0002          real*4 stat,contr,mslab,rmeas(1),rlab(1),evbuf(32),stbuf(32)
0003          real*4 buff(64),key(4),rnam1(4)
0004          integer*4 ndsg,ievr,istr,msrec,nxms,nstat,intst,nxst,icontr(12)
0005          integer*4 nret
0006          logical*1 temp1(4)
0007          equivalence (evbuf(31),nstat),(evbuf(32),intst), (stbuf(19),nxst),
             1            (buff(64),nxms), (stbuf(21),icontr)
0008          equivalence (rnam1(2),temp1)
0009          data rnam1/'sigm','    ','.dat',0./, star/'****'/
0010          common /flags/ iflgms,iflgev
0011          common /asvar/ nxrec1,nxrec2,nxrec3,nxrec4
      c
      c       construct data file name and open file (directory file should already be
      c             open)
0012          encode (4,1,temp1) ndsg
0013        1 format (i4)
0014          decode (4,2,temp1) temp1
0015          encode (4,2,temp1) temp1
0016        2 format (4i1)
0017          call setfil (2,rnam1)
0018          define file 2 (100,256,u,nxrec2)
      c
0019          nret=0
0020          key(1)=ndsg
0021          key(3)=contr
0022          if (iflgev.eq.0) call search (1,key,ievr,istr,icon,msrec,msind,
             1                              isrch)
0024          if (iflgev.eq.0.and.isrch.eq.1) go to 995
0026          if (stat.eq.star) go to 500
      c       search for specified station
0028          key(2)=stat
0029          call search(2,key,ievr,istr,icon,msrec,msind,isrch)
0030          if (isrch.eq.1) go to 995
      c       search for contractor index
0032          call search (3,key,ievr,istr,icon,msrec,msind,isrch)
0033          if (isrch.eq.1) go to 995
0035          if (iflgms.eq.1) go to 30
      c       return all measurements for station
```

```
UNIX fortran iv v01-11  source listing          page 002

0037          read (1'istr) stbuf
0038          ioff=icon-20
0039          msrec=icontr(ioff+1)
0040          nret=icontr(ioff+2)
0041          nblk=nret/31+1
0042          if (mod(nret,31).eq.0) nblk=nblk-1
0044          nxms=msrec
0045          do 10 n=1,nblk
0046          read (2'nxms) buff
0047          lim=nret-(n-1)*31
0048          if (lim.gt.31) lim=31
0050          do 20 k=1,lim
0051          ist=(n-1)*31+k
0052          rmeas(ist)=buff(k)
0053          rlab(ist)=buff(k+31)
0054       20 continue
0055       10 continue
0056          go to 995
0057       30 continue
0058          key(4)=mslab
0059          call search (4,key,ievr,istr,icon,msrec,msind,isrch)
0060          if (isrch.eq.1) go to 995
0062          read (2'msrec) buff
0063          nret=1
0064          rmeas(nret)=buff(msind)
0065          go to 995
0066      500 continue
      c       return one measurement for all stations across event
0067          key(4)=mslab
0068          read (1'ievr) evbuf
0069          nxst=intst
0070          do 40 k=1,nstat
0071          istr=nxst
0072          read (1'nxst) stbuf
      c       search for contractor index
0073          call search (3,key,ievr,istr,icon,msrec,msind,isrch)
0074          if (isrch.eq.1) go to 40
      c       search for measurement
0076          call search (4,key,ievr,istr,icon,msrec,msind,isrch)
0077          if (isrch.eq.1) go to 40
0079          read (2'msrec) buff
0080          nret=nret+1
0081          rmeas(nret)=buff(msind)
0082       40 continue
083       995 endfile 2
0084          return
0085          end
```

RETURN
(PAGE 1 OF 5)

```
UNIX fortran iv v01-11  source listing          page 001

0001         real*4 stat,contr,mslab,rmeas(100),rlab(100)
0002         integer*4 ndsg,nret,ievr
0003         data star/'****'/,ensc/'ensc'/
0004         data gumo/'gumo'/
0005         common /flags/ iflgms,iflgev
0006         common /asvar/ nxrec1,nxrec2,nxrec3,nxrec4
0007         call setfil (1,'sigms.dir')
0008         define file 1 (32000,128,u,nxrec1)
0009         iflgms=1
0010         write(6,100)
0011     100 format(' input event designation # (i4)')
0012         read(5,150) ndsg
0013     150 format(i4)
0014         write(6,200)
0015     200 format(' input station name (a4)')
0016         read(5,250) stat
0017     250 format(a4)
0018         if (stat.ne.star.and.iflgms.eq.0) go to 50
0020         write(6,300)
0021     300 format(' input measurement label (a4)')
0022         read(5,250) mslab
0023      50 continue
0024         contr=ensc
0025         iflgev=0
0026         call access (ndsg,stat,contr,mslab,nret,rmeas,rlab,ievr)
0027         write (6,10) nret,ievr
0028      10 format(5x,2i6)
0029         write (6,15) (rmeas(k),k=1,nret)
0030      15 format (1x,10f10.1)
0031         write (6,16) (rlab(k),k=1,nret)
0032      16 format (1x,10(a4,2x))
0033         stop
0034         end
```

```
UNIX fortran iv vØ1-11  source listing          page ØØ1

ØØØ1          subroutine search (iflag,key,evrec,strec,icon,msrec,msind,isrch)
       c      this routine searches for a specific keyword and returns the
       c      appropriate record and/or index
       c
       c      iflag=Ø     full search (needs all keys)
       c      iflag=1     search for event
       c      iflag=2     search for station (needs evrec)
       c      iflag=3     search for contractor (needs strec)
       c      iflag=4     search for measurement (needs icon)
       c         (returns rec # in 'sigmxxxx.dat' and index into rec)
       c      key(1)=event designation
       c      key(2)=station name
       c      key(3)=contractor name
       c      key(4)=measurement mnemonic
       c      isrch=Ø     successful search
       c      isrch=1     unsuccessful search
       c
ØØØ2          integer*4 nev,intev,ndr,mndr,evrec,strec,msrec,nmeas
ØØØ3            integer*4 ndsg,nxev,nstat,intst,nxst,nxms,icontr(12)
ØØØ4          real*4 key(4),buff(32),msbuff(64)
ØØØ5          equivalence (buff(1),ndsg,stat), (buff(19),nxst), (buff(3Ø),nxev)
ØØØ6          equivalence (buff(31),nstat), (buff(32),intst), (msbuff(64),nxms),
             1     (buff(21),icontr)
ØØ37          common /asvar/ nxrec1,nxrec2,nxrec3,nxrec4
ØØ38          isrch=Ø
ØØ39          if (iflag.eq.Ø) go to 1ØØ
ØØ11          go to (1ØØ,2ØØ,3ØØ,4ØØ), iflag
ØØ12       1ØØ read (1'1) nev,intev,ndr,mndr
ØØ13          if (nev.eq.Ø) go to 995
       c      search through events
ØØ15          nxev=intev
ØØ16          do 1Ø k=1,nev
ØØ17          evrec=nxev
ØØ18          read (1'nxev) buff
ØØ19          if (key(1).eq.float(ndsg)) go to 15
ØØ21       1Ø continue
ØØ22          write (6,14) key(1)
ØØ23       14 format('  event: ',f4.Ø,' not found')
ØØ24          isrch=1
ØØ25          go to 999
ØØ26       15 continue
ØØ27          if (iflag.eq.Ø) go to 2Ø1
ØØ29          return
ØØ3Ø       2ØØ continue
       c      search through stations given base vector (event) record #
ØØ31          if (evrec.eq.Ø) go to 995
ØØ33          read (1'evrec) buff
ØØ34       2Ø1 if (nstat.eq.Ø) go to 995
ØØ36          kount=nstat
ØØ37          nxst=intst
ØØ38          do 2Ø k=1,kount
ØØ39          strec=nxst
ØØ4Ø          read (1'nxst) buff
ØØ41          if (key(2).eq.stat) go to 25
ØØ43       2Ø continue
ØØ44          write (6,24) key(2)
ØØ45       24 format('  station: ',a4,' not found')
```

```
UNIX fortran iv v01-11  source listing          page 002

0046          isrch=1
0047          go to 999
0048       25 continue
0049          if (iflag.eq.0) go to 301
0051          return
0052      300 continue
      c       search station record for contractor index
0053          if (strec.eq.0) go to 995
0055          read (1'strec) buff
0056      301 continue
0057          do 30 icon=21,30,3
0058          if (key(3).eq.buff(icon)) go to 35
0060       30 continue
0061          write (6,34) key(3)
0062       34 format('  contractor: ',a4,' not found')
0063          isrch=1
0064          go to 999
0065       35 continue
0066          if (iflag.eq.0) go to 400
0068          return
0069      400 continue
0070          ioff=icon-20
0071          msrec=icontr(ioff+1)
0072          nmeas=icontr(ioff+2)
0073          if (nmeas.eq.0) go to 995
0075          nblk=nmeas/31+1
0076          if (mod(nmeas,31).eq.0) nblk=nblk-1
0078          nxms=msrec
0079          do 45 n=1,nblk
      c       set msrec to current rec #
0080          msrec=nxms
0081          read (2'nxms) msbuff
      c     search through measurement mnemonics
0082          do 40 k=32,62
0083          if (key(4).eq.msbuff(k)) go to 50
0085       40 continue
0086       45 continue
0087          write (6,46) key(4)
0088       46 format('  measurement: ',a4,' not found')
0089          isrch=1
0090          go to 999
0091       50 continue
0092          msind=k-31
0093          go to 999
0094      995 write (6,996)
0095      996 format('  error on search')
0096          stop
0097      999 return
0098          end
```

RETURN
(PAGE 4 OF 5)

```
UNIX fortran iv v01-11  source listing          page 001

0001          subroutine access (ndsg,stat,contr,mslab,nret,rmeas,rlab,ievr)
      c       this routine accesses the signal measurement data base
      c            ndsg=event designation #
      c            stat=station name (if = '****',given measurement is returned for all
      c                      stations for event)
      c            contr=contractor name
      c            iflgms=0  return all measurements for station (in rmeas) with assoc.
      c                      labels (in rlab)--------warning: it should be noted that
      c                      there is no guarantee all the meas. will be in the same
      c                      order across stations
      c            iflgms=1  return only indicated measurement
      c            mslab=label of desired measurement
      c            nret=# of measurements returned
      c            rmeas=measurement values
      c            rlab=associated measurement labels
      c            iflgev=0   search for event
      c            iflgev=1   use previous or supplied event record # (ievr)
      c            ievr=event record #
      c
0002          real*4 stat,contr,mslab,rmeas(1),rlab(1),evbuf(32),stbuf(32)
0003          real*4 buff(64),key(4),rnam1(4)
0004          integer*4 ndsg,ievr,istr,msrec,nxms,nstat,intst,nxst,icontr(12)
0005          integer*4 nret
0006          logical*1 temp1(4)
0007          equivalence (evbuf(31),nstat),(evbuf(32),intst), (stbuf(19),nxst),
             1            (buff(64),nxms), (stbuf(21),icontr)
0008          equivalence (rnam1(2),temp1)
0009          data rnam1/'sigm','    ','.dat',0./, star/'****'/
0010          common /flags/ iflgms,iflgev
0011          common /asvar/ nxrec1,nxrec2,nxrec3,nxrec4
      c
      c       construct data file name and open file (directory file should already be
      c            open)
0012          encode (4,1,temp1) ndsg
0013        1 format ('i4)
0014          decode (4,2,temp1) temp1
0015          encode (4,2,temp1) temp1
0016        2 format (4i1)
0017          call setfil (2,rnam1)
0018          define file 2 (100,256,u,nxrec2)
      c
0019          nret=0
0020          key(1)=ndsg
0021          key(3)=contr
0022          if (iflgev.eq.0) call search (1,key,ievr,istr,icon,msrec,msind,
             1                                  isrch)
0024          if (iflgev.eq.0.and.isrch.eq.1) go to 995
0026          if (stat.eq.star) go to 500
      c       search for specified station
0028          key(2)=stat
0029          call search(2,key,ievr,istr,icon,msrec,msind,isrch)
0030          if (isrch.eq.1) go to 995
      c       search for contractor index
0032          call search (3,key,ievr,istr,icon,msrec,msind,isrch)
0033          if (isrch.eq.1) go to 995
0035          if (iflgms.eq.1) go to 30
      c       return all measurements for station
```

RETURN
(PAGE 5 OF 5)

```
UNIX fortran iv v01-11  source listing          page 002

0037          read (1'istr) stbuf
0038          ioff=icon-20
0039          msrec=icontr(ioff+1)
0040          nret=icontr(ioff+2)
0041          nblk=nret/31+1
0042          if (mod(nret,31).eq.0) nblk=nblk-1
0044          nxms=msrec
0045          do 10 n=1,nblk
0046          read (2'nxms) buff
0047          lim=nret-(n-1)*31
0048          if (lim.gt.31) lim=31
0050          do 20 k=1,lim
0051          ist=(n-1)*31+k
0052          rmeas(ist)=buff(k)
0053          rlab(ist)=buff(k+31)
0054       20 continue
0055       10 continue
0056          go to 995
0057       30 continue
0058          key(4)=mslab
0059          call search (4,key,ievr,istr,icon,msrec,msind,isrch)
0060          if (isrch.eq.1) go to 995
0062          read (2'msrec) buff
0063          nret=1
0064          rmeas(nret)=buff(msind)
0065          go to 995
0066      500 continue
      c       return one measurement for all stations across event
0067          key(4)=mslab
0068          read (1'ievr) evbuf
0069          nxst=intst
0070          do 40 k=1,nstat
0071          istr=nxst
0072          read (1'nxst) stbuf
      c       search for contractor index
0073          call search (3,key,ievr,istr,icon,msrec,msind,isrch)
0074          if (isrch.eq.1) go to 40
      c       search for measurement
0076          call search (4,key,ievr,istr,icon,msrec,msind,isrch)
0077          if (isrch.eq.1) go to 40
0079          read (2'msrec) buff
0080          nret=nret+1
0081          rmeas(nret)=buff(msind)
0082       40 continue
0083      995 endfile 2
084           return
0385          end
```

APPENDIX C

FORMAT DESCRIPTION OF ENSCO'S RAW SIGNAL MEASUREMENT TAPE

FORMAT DESCRIPTION OF ENSCO'S RAW SIGNAL MEASUREMENT TAPE

The Raw Signal Measurement Tape is one of the tapes generated by the automated Signal Extraction Process of the Event Identification System (EIS) on the IBM 360/44 computer. This tape is a 1600 bpi nine-track tape and is written in the standard ENSCO format using physical I/O.

The standard nine-track tape format consists of a label file followed by one or more data files, followed by a trailer file. Each record in the data files represents a physical transfer of a specified number of bytes from program memory to magnetic tape. This results in efficient tape I/O and in a compact and easily interpreted tape. Figure C-1 illustrates the organization of the Raw Signal Measurement Tape. The individual components are described below.

The label file consists of two 80-byte records followed by an end-of-file mark (EOF). The first record is the IBM volume-serial header and begins with the characters VOL1. The second record is the ENSCO header and begins with the characters HDR1. The label file should be recognized by the IBM 360/44 as a standard tape label.

The data files each consist of a 1500-byte event header record followed by a 400-byte data record containing the raw signal measurements, followed by an end-of-file mark. A separate data file is generated for each event-station processed, and the data files are organized by site number (event header word 91). The stations corresponding to a given site number are presented in Table C-1 along with their locations and tectonic class.

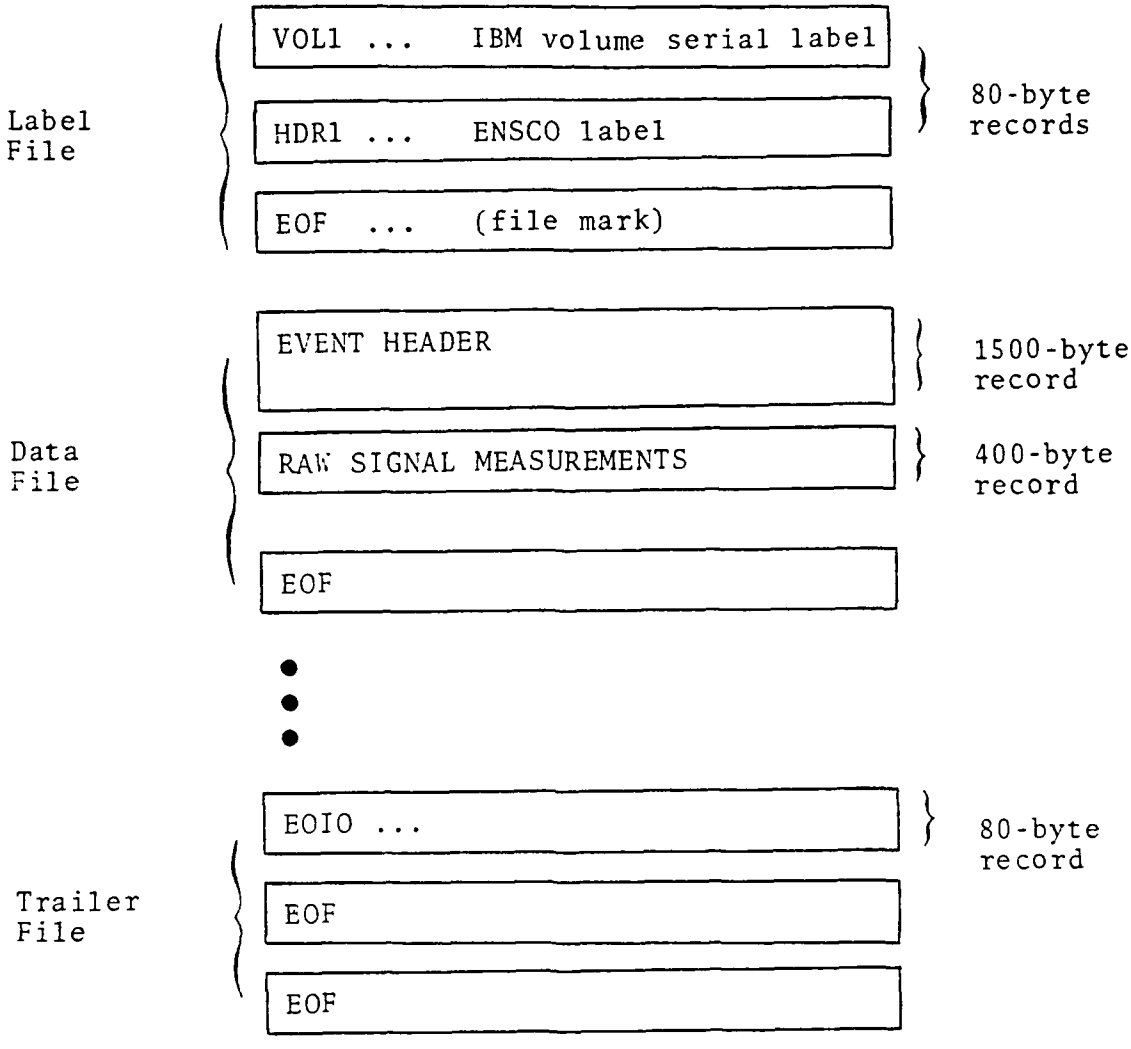


FIGURE C-1

ORGANIZATION OF RAW SIGNAL MEASUREMENT TAPE

TABLE C-1

SITE NUMBER ASSIGNMENTS

Site No.	Station	Latitude	Longitude	Tectonic Class
1	ANMO	34.95	-106.46	R
2	ANTO	39.87	32.79	A
3	BOCO	4.59	-74.04	A
4	CHTO	18.79	98.98	A
5	NORS	60.84	10.89	I
6	GUMO	13.59	144.87	A
7	MAIO	36.30	59.49	A
8	LASA	46.69	-106.22	I
9	NWAO	-32.93	117.24	I
10	GRFO	49.69	11.22	I
11	SHIO	25.57	91.88	A
12	TATO	24.98	121.49	A
13	SNZO	-41.31	174.70	A
14	ILPA	35.70	50.61	A
15	ALPA	65.00	-147.20	A
16	CTAO(ASRO)	-20.09	146.25	I
17	ZOBO	-16.27	-68.13	A
18	KAAO	34.54	69.04	A
19	MAJO	36.54	138.21	A
20	KONO	59.65	9.60	I
21	BFAK	64.77	-146.89	A
22	CTAO(HGLP)	-20.09	146.25	I
23	CHGO	18.79	98.98	A
24	TNAK	62.91	-156.02	A
25	TLOO	39.86	-4.01	I
26	EIAO	29.55	34.95	R
27	KONO	59.65	9.60	I
28	OGDO	41.07	-74.62	I
29	KIPO	21.42	-158.02	A
30	ALQO	34.94	-106.46	R
31	ZLPO	-16.27	-68.13	A
32	MATO	36.54	138.21	A
33	HNME	46.16	-67.99	I
34	RKON	50.84	-95.67	I
35	KSRS	37.45	127.92	I
36	ATAK	52.88	173.17	A
37	UCAK	66.00	-153.72	A
38	CNAK	67.45	-144.52	A
39	NJAK	63.06	-141.85	A
40-50	Unassigned	-----	------	-

Table C-2 describes all possible event header entries. In this application, not all the entries in the header record are used. Entries in the header corresponding to seismic information for specific phases should be disregarded. Information pertaining to an event or a station, or to non-seismic data (i.e., data record length, etc.) is still meaningful.

Table C-3 defines the raw signal measurements present in the data record. The amplitude measurements (words 1-21) are corrected for distance using appropriate B-factors (Sax et al., 1978). In addition, the data entries have either +1000 or -1000 added to them to indicate whether the value is a signal measurement or a noise measurement, respectively. A zero entry indicates that no measurement is available for the corresponding quantity. No regional S or Lg data were processed for this data base.

Finally, the data files are followed by a trailer file containing one 80-byte record beginning with the characters EOIO. This record is followed by at least two consecutive end-of-file marks.

TABLE C-2

STANDARD EVENT HEADER DESCRIPTION
(PAGE 1 OF 7)

Words	Data Type*	Field
1	I	Seismogram number.
2	I	Number of components (1 or 3).
3	I	Edit length (time points).
4	F	Number of samples per second.
5	F	Edit start time (hours).
6	F	Edit start time (minutes).
7	F	Edit start time (seconds).
8-9	A	Array name.
10-12	A	Event designation.
13-14	A	Data type ('EDIT', 'GEN', or 'DISCR').
15	I	Data record length (bytes).
16	I	Number of edited sites.
17-18	A	Data partition ('SIGNAL' or 'NOISE')
19	A	Domain ('TIME' or 'FREQ').
20-21	A	Source routine.
22	A	Data orientation code ('RAW' or 'PBD')
23	I	Maximum number of channels processed.
24	I	Number of channels deleted.
25	F	Re-sample rate.
26	I	Edit start time (seconds into day).
27	I	Edit start time (year-day).

*A = Alphanumeric, I = Integer, F = Floating Point

TABLE C-2

STANDARD EVENT HEADER DESCRIPTION
(PAGE 2 OF 7)

Words	Data Type*	Field
28	F	Edit length (seconds).
29-33	B	Site status table (0=present, 1=absent).
34	I	Number of uncorrectable data spikes encountered during edit.
35	I	Number of data clips encountered during edit.
36	I	Filter application code.
37	I	Smoothing code.
38	I	CH equalization code.
39	I	Calibration Code.
40	I	QC procedure code.
41	I	Taper code.
42	F	Initial beam or bandpass frequency.
43	F	Final beam or bandpass frequency.
44	F	Frequency increment.
45	I	Number of frequencies.
46	I	Length of transform (time points).
47	I	Length of data transformed, or integration gate (time points).
48	I	Number of transforms stacked.
49	I	Number of edited transforms deleted from stacking.
50	F	Noise taper coefficient.

*A = Alphanumeric, I = Integer, F = Floating Point

TABLE C-2

STANDARD EVENT HEADER DESCRIPTION
(PAGE 3 OF 7)

Words	Data Type*	Field
51	I	Index of signal start, relative to edit start.
52	I	Source time (seconds into day).
53	I	Source time (year-day).
54	A	Confidence of source time (PDE code).
55	F	Source latitude (± 90° N).
56	F	Source longitude (0-360° E).
57	F	Source depth (km).
58-59	A	Information source.
60	F	m_b.
61	F	M_s.
62	F	TI estimated M_s.
63	A	Standard deviation of residual time.
64	A	Number of stations in PDE reporting.
65-69	A	Seismic region.
70	A	NORSAR quality.
71	A	Sub-region.
72	I	P wave arrival time (seconds into day).
73	I	P wave arrival time (year-day).
74	I	S wave arrival time (seconds into day).
75	I	S wave arrival time (year-day).
76	I	LQ wave arrival time (seconds into day).
77	I	LQ wave arrival time (year-day).

*A = Alphanumeric, I = Integer, F = Floating Point

TABLE C-2

STANDARD EVENT HEADER DESCRIPTION
(PAGE 4 OF 7)

Words	Data Type*	Field
78	I	LR wave arrival time (seconds into day).
79	I	LR wave arrival time (year-day).
80	I	Estimated LR length (seconds).
81	F	Azimuth (Great Circle beam direction).
82	F	Elevation (Great Circle beam direction).
83	F	Azimuth (Primary beam direction).
84	F	Elevation (Primary beam direction).
85	F	Source to Array Great Circle distance (degrees).
86	F	Source to Array Great Circle distance (km).
87-88	A	Seismometer type.
89	A	Recording type.
90	F	Estimated signal-to-noise ratio (dB).
91	I	Site Number.
92	F	Site latitude (± 90° N).
93	F	Site longitude (0-360° E).
94	I	Number of sub-arrays.
95-134	I	Index of the first sensor in each sub-array.
135-174	F	Reference sensor latitude (±90°N) for each sub-array.
175-214	F	Reference sensor longitude (0-360°E) for each sub-array.
215	A	Tectonic class code.

*A = Alphanumeric, I = Integer, F = Floating Point

TABLE C-2

STANDARD EVENT HEADER DESCRIPTION
(PAGE 5 OF 7)

Words	Data Type*	Field	
		Long-Period	Short-Period
216	F	Log_{10} of 50 second amplitude (vertical component).	S wave, LG (surface) wave, or regional P wave magnitude.
217	F	Log_{10} of 33 second amplitude (vertical component).	Teleseismic P wave magnitude.
218	F	Log_{10} of 25 second amplitude (vertical component).	Measured phase arrival time (seconds into edit) Value of 999999 indicates no detection.
219	F	Log_{10} of 20 second amplitude (vertical component).	Magnitude, from first envelope peak.
220	F	Log_{10} of 17 second amplitude (vertical component).	Mean smoothed frequency.
221	F	Log_{10} of 14 second amplitude (vertical component).	Maximum mean smoothed frequency.
222	F	Log_{10} of 12 second amplitude (vertical component).	Mean phase standard deviation.
223	F	Log_{10} of 50 second amplitude (transverse component).	Log_{10} of center frequency no. 1.
224	F	Log_{10} of 33 second amplitude (transverse component).	Log_{10} of center frequency no. 2.

*A = Alphanumeric, I = Integer, F = Floating Point

TABLE C-2

STANDARD EVENT HEADER DESCRIPTION
(PAGE 6 OF 7)

Words	Data Type*	Field	
		Long-Period	Short-Period
225	F	Log_{10} of 25 second amplitude (transverse component).	Log_{10} of center frequency no. 3.
226	F	Log_{10} of 20 second amplitude (transverse component).	Log_{10} of center frequency no. 4.
227	F	Log_{10} of 17 second amplitude (transverse component).	Log_{10} of center frequency no. 5.
228	F	Log_{10} of 14 second amplitude (transverse component).	Log_{10} of center frequency no. 6.
229	F	Log_{10} of 12 second amplitude (transverse component).	Log_{10} of center frequency no. 7.
230	F	Log_{10} of broadband A/T.	Log_{10} of center frequency no. 8.
231-238	F	Undefined.	Log_{10} of displacement at center frequencies no.1-8.
239	F	Undefined.	Broadband complexity.
240	F	Undefined.	Envelope complexity.
241-248	F	Undefined.	Instantaneous frequency complexity at center frequencies no. 1-8.
249	F	Undefined.	m_p (taken from first five seconds of data.

*A = Alphanumeric, I = Integer, F = Floating Point

TABLE C-2

STANDARD EVENT HEADER DESCRIPTION
(PAGE 7 OF 7)

Words	Data Type*	Field	
		Long-Period	Short-Period
250	F	Undefined.	Lg arrival time (seconds into day).
251-265	F	Variable-frequency detection ratios.	
266-267	A	Date on which data were edited.	
268	I	Start time of corresponding seismogram on a subset tape (not used for AEDS data).	
269	F	B factor for P_n wave for source-receiver distance (AEDS data, only).	
270	F	B factor for P wave for source-receiver distance (AEDS data, only).	
271	F	B factor for Sn wave for source-receiver distance (AEDS data, only).	
272	F	B factor for lg wave for source-receiver distance (AEDS data, only).	
273	F	B factor for long-period surface wave for source-receiver distance (AEDS data, only)	
274	I	Edit start time index into the available data (AEDS data, only).	
275-276		Available for future use.	
277-336	I	Parameters used by subroutine MSDISC (AEDS data, only).	
296-335	F	Sensor East Cartesian coordinates with respect to the reference sensor (km).	
336-375	F	Sensor North Cartesian coordinates with respect to the reference sensor (km).	

*A = Alphanumeric, I = Integer, F = Floating Point

TABLE C-3

RAW SIGNAL MEASUREMENTS

Real *4 Word	Description
	Long-Period
1	18-22 sec Vertical Log A/T *
2	50 sec Vertical Log A **
3	33.3 sec Vertical Log A
4	25 sec Vertical Log A
5	20 sec Vertical Log A
6	17 sec Vertical Log A
7	14 sec Vertical Log A
8	12 sec Vertical Log A
9	25 sec Transverse Log A
	Short-Period
10	P-wave log A/T ($\Delta<20^{o}$)
11	S-wave log A/T ($\Delta<20^{o}$)
12	Lg-wave log A/T ($\Delta<20^{o}$)
13	P-wave log A/T ($\Delta>20^{o}$)
14	0.316 Hz Vertical Log A
15	0.501 Hz Vertical Log A
16	0.794 Hz Vertical Log A
17	1.259 Hz Vertical Log A
18	1.995 Hz Vertical Log A
19	3.162 Hz Vertical Log A
20	5.012 Hz Vertical Log A
21	7.943 Hz Vertical Log A
22	Maximum mean frequency
23	Mean phase standard deviation
24	Broadband complexity
25	Minimum narrowband complexity

* Log A/T = logarithm (base 10) amplitude/period + B-factor

** Log A = logarithm (base 10) amplitude + B-factor

REFERENCES

Sax, R. L., and Technical Staff, 1978; Event Identification - Applications to Area of Interest Events, Technical Report No. 20, Texas Instruments Report No. ALEX(01)-TR-78-08, AFTAC Contract Number F08606-77-C-0004, Texas Instruments Incorporated, Dallas, TX.